(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 July 2003 (24.07.2003)

PCT

(10) International Publication Number
WO 03/061188 A1

(51) International Patent Classification[7]: H04L 9/00

(21) International Application Number: PCT/US03/00817

(22) International Filing Date: 13 January 2003 (13.01.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/347,243 14 January 2002 (14.01.2002) US

(71) Applicant *(for all designated States except US)*: **NETMOTION WIRELESS, INC.** [US/US]; 1100 Dexter Avenue North, Seattle, WA 98109 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **STURNIOLO, Emil** [US/US]; NetMotion Wireless, Inc., 1100 Dexter Avenue N., Seattle, WA 98109 (US). **STAVENS, Aaron** [US/US]; NetMotion Wireless, Inc., 1100 Dexter Avenue N., Seattle, WA 98109 (US). **SAVARESE, Joseph, T.** [US/US]; NetMotion Wireless, Inc., 1100 Dexter Avenue N., Seattle, WA 98109 (US).

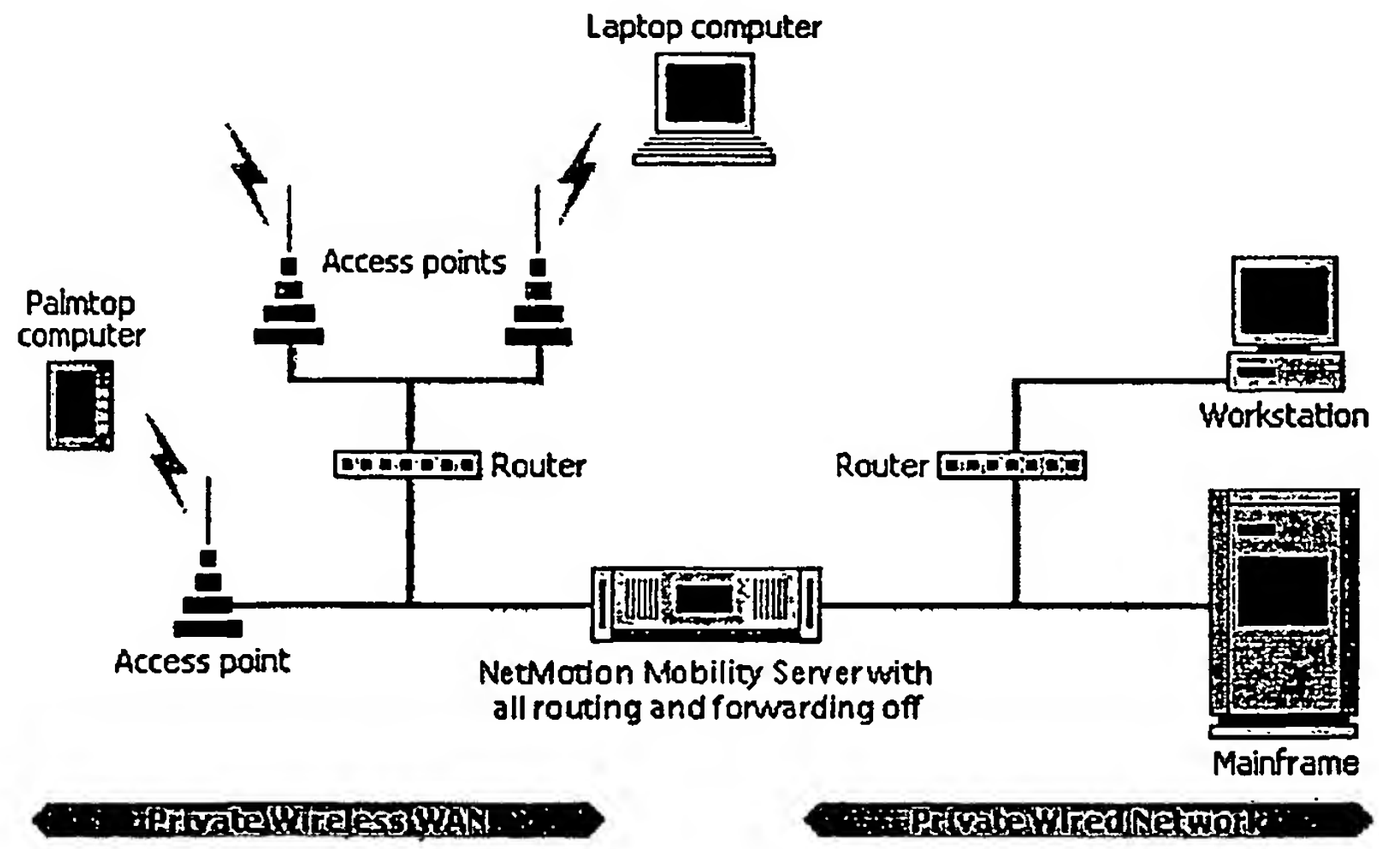
(74) Agent: **FARIS, Robert, W.**; Nixon & Vanderhye P.C., 1100 North Glebe Road, Suite 800, Arlington, VA 22201-4714 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE,

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR PROVIDING SECURE CONNECTIVITY IN MOBILE AND OTHER INTERMITTENT COMPUTING ENVIRONMENTS



(57) Abstract: Method and apparatus for enabling secure connectivity using standardsbased Virtual Private Network (VPN) IPSEC algorithms in a mobile and intermittently connected computing environment enhance the current standards based algorithms by allowing migratory devices to automatically (re)establish security sessions as the mobile end system roams across homogeneous or heterogeneous networks while maintaining network application session. The transitions between and among networks occur seamlessly -- shielding networked applications from interruptions in connectivity. The applications and/or users need not be aware of these transitions, although intervention is possible. The method does not require modification to existing network infrastructure and/or modification to networked applications.

WO 03/061188 A1

ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Published:

— *with international search report*

TITLE OF THE INVENTION

METHOD AND APPARATUS FOR PROVIDING SECURE CONNECTIVITY IN MOBILE AND OTHER INTERMITTENT COMPUTING ENVIRONMENTS

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of priority from the following copending commonly-assigned related US patent applications:

U.S. Provisional Application No. 60/347,243, filed January 14, 2002 (Attorney Docket 3978-9);

U.S. Provisional Application Serial No. 60/274,615 filed March 12, 2001, entitled "Method And Apparatus For Providing Mobile and Other Intermittent Connectivity In A Computing Environment" (Attorney Docket 3978-6);

U.S. Patent Application Serial No. 09/330,310 filed June 11, 1999, entitled "Method And Apparatus For Providing Mobile and Other Intermittent Connectivity In A Computing Environment" (Attorney Docket 3978-3);

U.S. Patent Application Serial No. 09/660,500 filed September 12, 2000, entitled "Method And Apparatus For Providing Mobile and Other Intermittent Connectivity In A Computing Environment" (Attorney Docket 3978-2); and

PCT International Application Number PCT/US01/28391 filed September 12, 2001, entitled "Method And Apparatus For Providing Mobile And Other Intermittent Connectivity In A Computing Environment" (Attorney Docket 3978-7).

All of the above-identified documents are incorporated herein by reference.

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

Not applicable.

BACKGROUND AND SUMMARY OF THE INVENTION

Wireless networks have become very popular. Students are accessing course information from the college's computer network while sitting in lecture hall or enjoying the outdoors in the middle of the college campus. Doctors are maintaining computing connectivity with the hospital computer network while making their rounds. Office workers can continue to work on documents and access their email as they move from their office to a conference room. Laptop or PDA users in conference centers, hotels, airports and coffee houses can surf the web and access email and other applications over the Internet. Home users are using wireless networks to eliminate the need to run cables.

Wireless connectivity provides great flexibility but also presents security risks. Information transmitted through a cable or other wired network is generally secure because one must tap into the cable in order to access the transmission. However, information transmitted wirelessly can be received by anyone with a wireless receiver who is in range. Security risks may not present much of a problem to students reading course material or to café customers surfing the World Wide Web, but they present major concerns to businesses and professionals as well as their clients, customers and patients.

Generally, wired and wireless computing worlds operate under very different paradigms. The wired world assumes a fixed address and a constant connection with high bandwidth. A wireless environment, in contrast, exhibits intermittent connections and has higher error rates over what is usually a narrower bandwidth. As a result, applications and messaging protocols designed for the wired world don't always work in a wireless environment. However, the wireless expectations of end users are set by the performance and behaviors of their wired networks. Meeting these expectations creates a

significant challenge to those who design and develop wireless networking architectures, software and devices.

Authenticating users and keeping communications confidential are more problematic in a wireless network than they are in a wired network. Wireless networks generally are subject to much greater varieties of attacks (e.g., man-in-the-middle, eavesdropping, "free rides" and wide area imposed threats) and assumptions that often do not apply to wired networks. For example, in modern network topologies such as wireless networks and Internet-based virtual private networks (VPNs), physical boundaries between public and private networks do not exist. In such networks, whether a user has the necessary permissions to access the system can no longer be assumed based on physical location as with a wired network in a secure facility. Additionally, wireless data is often broadcasted on radio frequencies, which can travel beyond the control of an organization, through walls and ceilings and even out into the parking lot or onto the street. The information the network is carrying is therefore susceptible to eavesdropping. Imagine if vital hospital patient information could be intercepted or even altered by an unauthorized person using a laptop computer in the hospital lobby, or if a corporate spy could learn his competitor's secrets by intercepting wireless transmissions from an office on the floor above or from a car in the parking lot. While tapping into a wired network cable in a secure facility is possible, the chances of this actually happening are less likely than interception of radio transmissions from a wireless network. Further security threats and problems must be faced when users wish to use any of the ever-increasing variety of public wireless networks to access sensitive data and applications.

Many of the open standards that make it possible for wireless network hardware vendors to create interoperable systems provide some form of security protection. For example, the IEEE 802.11b "Wi-Fi" standard has been widely implemented to provide wireless connectivity for all sorts of computing devices. It provides an optional Wired Equivalent Privacy ("WEP") functionality that has been widely implemented. Various additional wireless

related standards attempt to address security problems in wireless networks, including for example:

- Wireless Application Protocol (WAP) and the associated Wired Transport Layer Security (WTLS); and
- Mobile IP.

However, as explained below and as recognized throughout the industry, so far these standards have not provided a complete, easy-to-implement transparent security solution for mobile computing devices that roam between different networks or subnetworks.

WAP generally is designed to transmit data over low-bandwidth wireless networks to devices like mobile telephones, pagers, PDA's, and the like. The Wired Transport Layer Security (WTLS) protocol in WAP provides privacy, data integrity and authentication between WAP-based applications. A WAP gateway converts between the WAP protocol and standard web and/or Internet protocols such as HTTP and TCP/IP, and WTLS is used to create a secure, encrypted pipe. One issue with this model is that once the intermediate WAP gateway decrypts the data, it is available in clear text form -- presenting an opportunity for the end-to-end security of the system to be compromised. Additionally, WAP has typically not been implemented for high-bandwidth scenarios such as wireless local area network personal computer connectivity.

WEP (Wired Equivalent Privacy) has the goal of providing a level of privacy that is equivalent to that of an unsecured wired local area network. WEP is an optional part of the IEEE 802.11 standard, but many hardware vendors have implemented WEP. WEP provides some degree of authentication and confidentiality, but also has some drawbacks and limitations.

To provide authentication and confidentiality, WEP generally relies on a default set of encryption keys that are shared between wireless devices (e.g., laptop computers with wireless LAN adapters) and wireless access points. Using WEP, a client with the correct encryption key can "unlock" the network and communicate with any access point on the wireless

network; without the right key, however, the network rejects the link-level connection request. If they are configured to do so, WEP-enabled wireless devices and access points will also encrypt data before transmitting it, and an integrity check ensures that packets are not modified in transit. Without the correct key, the transmitted data cannot be decrypted – preventing other wireless devices from eavesdropping.

WEP is generally effective to protect the wireless link itself although some industry analysts have questioned the strength of the encryption that WEP currently uses. However, a major limitation of WEP is that the protection it offers does not extend beyond the wireless link itself. WEP generally offers no end-to-end protection once the data has been received by a wireless access point and needs to be forwarded to some other network destination. When data reaches the network access point or gateway, it is unencrypted and unprotected. Some additional security solution must generally be used to provide end-to-end authentication and privacy.

Mobile IP is another standard that attempts to solve some of the problems of wireless and other intermittently-connected networks. Generally, Mobile IP is a standards based algorithm that enables a mobile device to migrate its network point of attachment across homogeneous and heterogeneous network environments. Briefly, this Internet Standard specifies protocol enhancements that allow routing of Internet Protocol (IP) datagrams (e.g., messages) to mobile nodes in the Internet. See for example Perkins, C., "IP Mobility Support", RFC 2002, October 1996.

Mobile IP contemplates that each mobile node is always identified by its home address, regardless of its current point of attachment to the Internet. While situated away from its home, a mobile node is also associated with a "care-of" address, which provides information about its current point of attachment to the Internet. The protocol provides for registering the "care-of" address with a home agent. The home agent sends datagrams destined for the mobile node through a "tunnel" to the "care-of"

address. After arriving at the end of the "tunnel," each datagram is then delivered to the mobile node.

While Mobile IP provides useful techniques for remote connectivity, it is not yet widely deployed/implemented. This seems to be due to a variety of factors – at least one of which is that there continues to be some unsolved problems or areas where the Mobile IP standard is lacking and further enhancement or improvement would be desirable. For example, even though security is now fairly widely recognized as being a very important aspect of mobile networking, the security components of Mobile IP are still mostly directed to a limited array of security problems such as redirection attacks.

Redirection attacks are a very real threat in any mobility system. For example, a redirection attack can occur when a malicious node gives false information to a home agent in a Mobile IP network (e.g., sometimes by simply replaying a previous message). This is similar to someone filing a false "change of address" form with the Post Office so that all your mail goes to someone else's mailbox. The home agent is informed that the mobile node has a new "care-of" address. However, in reality, this new "care-of" address is controlled by the malicious node. After this false registration occurs, all IP datagrams addressed to the mobile node are redirected to the malicious node.

While Mobile IP provides a mechanism to prevent redirection attacks, there are other significant security threats that need to be addressed before an enterprise can feel comfortable with the security of their wireless network solution. For example, Mobile IP generally does not provide a comprehensive security solution including mobile computing capabilities such as:

- Session resilience/persistence
- Policy management
- Distributed firewall functionality
- Location based services
- Power management
- Other capabilities.

While much security work has been done by the Internet community to date in the Mobile IP and other contexts, better solutions are still possible and desirable. In particular, there continues to be a need to provide an easy-to-use, comprehensive mobility solution for enterprises and other organizations who wish to add end-to-end security to existing and new infrastructures that make extensive use of existing conventional technology and standards and which support mobility including roaming transparently to applications that may not be "mobile-aware." Some solutions exist, but many of them require changes to existing infrastructure that can be difficult to implement and maintain.

For example, in terms of the current implementations that do exist, Mobile IP is sometimes implemented as a "bump" in the TCP/IP protocol stack to replace components of the existing operating system environment. An example of such an architecture is shown in prior art Figure 1. In the exemplary illustrative prior art arrangement shown, a Mobile IP module sits below the regular TCP/IP protocol stack components and manages the transitions from one network to another. Generally, using such solution, additions or modifications to existing core network infrastructure entities are needed to facilitate the behavior of nomadic or migratory computing. The need for such modifications makes widespread implementation difficult and causes problems in terms of maintainability and compatibility.

Another common security solution that enterprises have gravitated toward is something called a Virtual Private Network (VPN). VPNs are common on both wired and wireless networks. Generally, they connect network components and resources through a secure protocol tunnel so that devices connected to separate networks appear to share a common, private backbone. VPN's accomplish this by allowing the user to "tunnel" through the wireless network or other public network in such a way that the "tunnel" participants enjoy at least the same level of confidentiality and features as when they are attached to a private wired network. Before a "tunnel" can be established, cryptographic methods are used to establish and authenticate the

identity of the tunnel participants. For the duration of the VPN connection, information traversing the tunnel can be encrypted to provide privacy.

VPN's provide an end-to-end security overlay for two nodes communicating over an insecure network or networks. VPN functionality at each node supplies additional authentication and privacy in case other network security is breached or does not exist. VPN's have been widely adopted in a variety of network contexts such as for example allowing a user to connect to his or her office local area network via an insecure home Internet connection. Such solutions can offer strong encryption such as the AES(Advanced Encryption Standard), compression, and link optimizations to reduce protocol chattiness. However, many or most VPNs do not let users roam between subnets or networks without "breaking" the secure tunnel. Also, many or most VPNs do not permit transport, security and application sessions to remain established during roaming. Another potential stumbling block is conventional operating systems – not all of which are compatible with the protection of existing wireless VPNs.

To address some of the roaming issue, as previously mentioned, standards efforts have defined Mobile IP. However, Mobile IP, for example, operates at the network layer and therefore does not generally provide for session persistence/resilience. If the mobile node is out of range or suspended for a reasonably short period of time, it is likely that established network sessions will be dropped. This can present severe problems in terms of usability and productivity. Session persistence is desirable since it lets the user keep the established session and VPN tunnel connected – even if a coverage hole is entered during an application transaction. Industry analysts and the Wireless Ethernet Compatibility Alliance recommend that enterprises deploy VPN technology, which directly addresses the security problem, and also provides advanced features like network and subnet roaming, session persistence for intermittent connections, and battery life management for mobile devices. However, VPN solutions should desirably support standard security encryption algorithms and wireless optimizations suitable for today's

smaller wireless devices, and should desirably also require no or minimal modification to existing infrastructure.

One standards-based security architecture and protocol approach that has been adopted for providing end-to-end secure communications is called "Internet Security Protocol" ("IPSec"). IPSec is a collection of open standards developed by the Internet Engineering Task Force (IETF) to secure communications over public and private networks. See for example:

- RFC 1827 "IP Encapsulating Security Payload (ESP)" R. Atkinson (August 1995);
- RFC 1826 "IP Authentication Header" R. Atkinson. (August 1995); and
- RFC 1825 "Security Architecture for the Internet Protocol" R. Atkinson (August 1995).

Briefly, IPSec is a framework for ensuring private, secure communications over Internet Protocol (IP) networks, through the use of cryptographic security services. The IPSec suite of cryptography-based protection services and security protocols provides computer-level user and message authentication, as well as data encryption, data integrity checks, and message confidentiality. IPSec capabilities include cryptographic key exchange and management, message header authentication, hash message authentication, an encapsulating security payload protocol, Triple Data Encryption, the Advanced Encryption Standard, and other features. In more detail, IPSec provides a transport mode that encrypts message payload, and also provides a tunnel mode that encrypts the payload, the header and the routing information for each message. To reduce overhead, IPSec uses policy-based administration. IPSec policies, rather than application programming interfaces (APIs), are used to configure IPSec security services. The policies provide variable levels of protection for most traffic types in most existing networks. One can configure IPSec policies to meet the security requirements of a computer, application, organizational unit, domain, site, or global enterprise based on IP address and/or port number.

IPSec is commonly used in firewalls, authentication products and VPNs. Additionally, Microsoft has implemented IPSec as part of its Windows 2000 and Windows XP operating system. IPSec's tunnel mode is especially useful in creating secure end-to-end VPNs. IPSec VPNs based on public key cryptography provide secure end-to-end message authentication and privacy. IPSec endpoints act as databases that manage and distribute cryptographic keys and security associations. Properly implemented, IPSec can provide private channels for exchanging vulnerable data such as email, file downloads, news feeds, medical records, multimedia, or any other type of information.

One might initially expect that it should be relatively straightforward to add a security algorithm such as the standards-based IPSec security algorithm to Mobile IP or other mobility protocol. For example, layering each of the entities in the fashion such as that shown in prior art Figure 2 would seem to allow for security in an environment where the mobile node's IP address never needs to change. Thus, the IPSec security association between the mobile node and its ultimate peer could be preserved across network segment boundaries, and end-to-end security would also preserved. However, combining the Mobile IP and IPSec algorithms in this manner can present its own set of problems.

For example, when the mobile node has roamed to a foreign network and is communicating with its ultimate peer, it is possible that packets generated by the mobile node may be discarded by a policy enforcement entity such as a firewall. This can be due to common practice known as ingress filtering rules. Many firewalls discard packets generated by mobile nodes using their home addresses (internal network identity) and received on an externally facing network interface in defense of the network. This discarding process is intended to protect the network secured by the firewall from being attacked. Ingress filtering has the effect of forcing the tunneling of Mobile IP frames in both directions. See for example RFC 2356 Sun's SKIP Firewall Traversal for Mobile IP: G. Montenegro, V. Gupta. (June 1998).

Additionally, it is becoming general practice in the industry to require that an IPSec security session be established between the foreign agent and the externally facing policy enforcement equipment (e.g. firewall) before allowing packets to traverse between the external and internal network interconnection (a.k.a. VPN). If the foreign agent is co-located with the mobile node, this can become a cumbersome operation. As exemplary Figure 3 depicts, yet another level of network protocol enveloping could be used to meet possibly required security policies to allow network traffic to flow between the mobile node to the foreign agent through the policy enforcement equipment (e.g.. firewall) to the home agent and then to the other communications end point (i.e. ultimate peer). However, this adds substantial additional overhead due to the additional encapsulation. Furthermore, if the foreign agent entity is not co-located with the mobile node (or up to policy restrictions on the newly attached network), a specific foreign agent may need to be used for these communications, and credential information must somehow be shared between the foreign agent and the terminus of the first (outer) IPSec session. A drawback to this methodology is that it can increase the security risk by sharing credential information with a network entity that may not be directly under user or corporate administrative control.

What is needed is a solution to these problems providing security, network roaming, and session persistence over conventional information communications networks including but not limited to standard IP based networks without requiring modification to existing network applications. Additionally, it would be useful if such a solution did not require the deployment of Mobile IP or any additional infrastructure such as a foreign agent when visiting a remote network, and the functionality can be transparent to networked applications so they do not need to be modified either.

This invention solves this problem by transparently providing secure, persistent, roamable IP-based communications using conventional technologies such as IPSec, Microsoft or other operating system security functionality while avoiding the commonly experienced ingress filtering

problems. And unlike at least some implementations of Mobile IP, few if any changes are necessary to the underlying network infrastructure.

Generally, one preferred exemplary non-limiting embodiment provides Mobility Client (MC) functionality that virtualizes the underlying network. Applications running on the mobility client see at least one consistent virtual network identity (e.g. IP address). When an application on the mobility client makes a network request, the mobility client intercepts the request and marshals the request to a Mobility Server (MS) that supports security such as IPSEC. The mobility server unwraps the request and places it on the network as though the server were the client – thus acting as a proxy for the client.

The reverse also occurs in the exemplary embodiment. When a peer host sends a packet to the mobility client's virtual network identity, the packet is first received by the mobility server and is then transferred to the mobility client. The mobility server maintains a stable point of communication for the peer hosts while the mobility client is free to roam among networks as well as suspend or roam out of range of any network. When the mobility client is out of range, the mobility server keeps the mobility client's sessions alive and queues requests for the mobility client. When the mobility client is once again reachable, the mobility server and client transfer any queued data and communication can resume where it left off..

Preferred exemplary non-limiting implementations thus offer wireless optimizations and network and application session persistence in the context of a secure VPN or other connection. Wireless optimizations allow data to be transmitted as efficiently as possible to make maximal use of existing bandwidth. For example, the system can be used to switch automatically to the fastest bandwidth network connection when multiple connections (Wi-Fi and GPRS, for example) are active. Network session persistence means that users don't have to repeat the login process when they move from one IP subnet to another, or when they go out of range of the network and return. Exemplary implementations automatically re-authenticate the connection every time users roam, without need for user intervention.

Application session persistence means that standard network applications remain connected to their peers, preventing the loss of valuable user time and data. Such optimizations and persistence is provided in the context of a security architecture providing end-to-end security for authentication and privacy.

In one illustrative embodiment, before data is transported between the network and a mobility client, the network ensures that the end user has the required permissions. A user establishes her identity by logging in to the mobility client using a conventional (e.g., Windows) domain user name and password. Using the conventional domain credentials allows for a single sign-on process and requires no additional authentication tables or other infrastructure additions. Single sign-on also gives users access to other domain resources such as file system shares. Once a user has been authenticated, a communications path is established for transporting application data. Any number of different protocols (e.g., Common Internet File System, Radius, other) can be used for user authentication. Using certain of these protocols, a mobility server can act as a Network Access Server to secure an initial access negotiation which establishes the user's user name and password using conventional protocols such as EAP-MD5, LEAP, or other protocol. Unlike some wireless protocols, such authentication in the exemplary non-limiting implementations provides user-specific passwords that can be used for policy management allowing access and resource allocation on a user basis.

Significantly, exemplary non-limiting implementations can be easily integrated with IPSEC or other security features in conventional operating systems such as for example Windows NT and Windows 2000. This allows access to conventional VPN and/or other proven-secure connection technology. IPSec policies can be assigned through the group policy feature of Active Directory, for example. This allows IPSec policy to be assigned at the domain or organizational level – reducing the administrative overhead of configuring each computer individually. An on-demand security negotiation and automatic key management service can also be provided using the

conventional IETF-defined Internet Key Exchange (IKE) as specified in Internet RFC 2409. Such exemplary implementations can provide IEFT standards-based authentication methods to establish trust relationships between computers using public key cryptography based certificates and/or passwords such as preshared keys. Integration with conventional standards-based security features such as public key infrastructure gives access to a variety of security solutions including secure mail, secure web sites, secure web communications, smart card logon processes, IPSec client authentication, and others.

Illustrative exemplary embodiments can be cognizant of changes in network identity, and can selectively manage transition in network connectivity, possibly resulting in the termination and/or (re)instantiation of IPSec security sessions between communicating entities over at least one of a plurality of network interfaces. Exemplary illustrative embodiments also provide for the central management, distribution, and/or execution of policy rules for the establishment and/or termination of IP security sessions as well as other parameters governing the behavior for granting, denying and/or delaying the consumption of network resources.

Illustrative non-limiting advantageous features include:

- Roamable IPSec allows IPSec tunnel to automatically roam with mobile computing devices wherever they go – based on recognized IPSec security standard, Roamable IPSec enables seamless roaming across any physical or electronic boundary with the authentication, integrity and encryption of IPSec, to provide a standards-based solution allowing mobile and remote users with VPN-level security and encryption in an IPSec tunnel that seamlessly roams with wireless users wherever they go and however they access their enterprise data.
- Detecting when a change in network point of attachment, an interruption of network connectivity, a roam to a different network or other subnetworks, a mobile client's identify, or other discontinuity has occurred on the mobile client and (re)instantiating an IP Security session while maintaining

network application sessions -- all in a manner that is transparent to the networked application.

- Transparently and selectively injecting computer instructions and redirecting the execution path of at least one software or other component based for example on process name to achieve additional level(s) of functionality while maintaining binary compatibility with operating system components, transport protocol engines, and/or applications.
- Selectively but transparently virtualizing at least one network interface for applications and operating system components -- shielding them from the characteristics of mobile computing while allowing other components to remain cognizant of interruptions in connectivity and changes in network point of attachment.
- Selectively virtualizing at least one network interface for network applications and operating system components thus shielding them from adverse events that may disturb communications such as changes in network point of attachment and/or periods of disconnectedness.
- Allowing the establishment of multiple IP Security sessions over one or more network interfaces associated with at least one network point of attachment and allowing network application communications to simultaneously flow over any or all of the multiple IP security sessions and correctly multiplex/demultiplex these distributed communication flows into corresponding higher layer communications sessions.
- Applying policy rules to selectively allow, deny, and/or delay the flow of network communications over at least one of a plurality of IP Security sessions.
- Centrally managing and/or distributing policy regarding the establishment of IP Security sessions from a central authority.
- An "Add session" concept -- during the proxying of communications for a mobile client, the mobility server can instantiate at least one of a possible plurality of IP Security sessions between a mobility server and an ultimate peer on behalf of a mobility client.

- Establishing and maintaining IP Security sessions between the Mobility Server and ultimate communications peer, even during periods when the mobility client is unreachable.
- Automatically terminating IP Security sessions between the mobility server and ultimate communications peer, based on, but not limited to link inactivity, application session inactivity, or termination of a communications end point.
- Associating at least one IP security session between the mobility server and ultimate peer and mobility client and mobility server regardless of the current mobility client network identities.
- Transparently injecting computer instructions and redirecting the execution path to achieve additional level(s) of functionality while maintaining binary compatibility with operating system components, transport protocol engines, and applications.
- Allowing establishment of at least one of a plurality of IP security sessions over a plurality of network interfaces associated with at least one network point of attachment and allowing network application communications to simultaneously flow over at least one of a plurality of IP Security sessions and correctly multiplex/demultiplex these distributed communication flows back into corresponding higher layer communications sessions.
- Selectively virtualizing at least one network interface for network applications and operating system components thus shielding them any adverse events that may disrupt communications such as changes in network point of attachment and/or periods of disconnectedness.
- Centrally managing and distributing policy rules regarding the establishment and/or termination of IP security sessions for mobile clients and/or mobility servers from a central authority.
- A mobility security solution that starts at the mobile device and provides both secure user authentication and, when needed, secure data encryption.
- A mobility security solution that voids the need for single-vendor solutions not based on industry-wide, open and other standards.

- Secure VPN that is extendable to a variety of different public data networks having different configurations (e.g., Wi-Fi network hotspot, wide-area wireless solutions such as CDPD or GPRS, etc.) dynamically controllable by the network administrator
- A mobility security solution that works with a wide variety of different computing devices of different configurations running different operating systems.
- Allows users to suspend and reestablish secure sessions to conserve battery power while maintaining network application sessions.
- Provides a secure solution in a wireless topology that has dead spots and coverage holes.
- No need to develop custom mobile applications or use mobile libraries to get applications to work in a mobile environment.
- Secure transport and application session persistence
- works within existing network security so the network is not compromised.
- compatible with any of a variety of conventional security protocols including for example RADIUS, Kerberos, Public Key Infrastructure (PKI), and Internet Security Protocol (IPSec).
- The computing environment and the applications do not need to change – mobility is there to use but its use is transparent to the user and to the applications.
- Since all or nearly all applications run unmodified, neither re-development nor user re-training is required.
- Automatic regeneration of user-session keys at a customized interval.
- Continuous, secure connection ensuring data integrity between wired and wireless data networks.
- Enterprises running VPNs (e.g., PPTP, L2TP/IPSec, IPSec, Nortel, Cisco, other) can use these techniques to add wireless optimization, session persistence and additional security for mobile workers.

- Seamlessly integrates into enterprises where LEAP or other access point authentication security is deployed to add optimized roamable security and encryption.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other features and advantages may be better and more completely understood by referring to the following detailed description of exemplary non-limiting illustrative embodiments in conjunction with drawings, of which:

Figure 1 shows an exemplary illustrative prior art mobile IP client architecture;

Figure 2 shows an exemplary illustrative prior art IPSec and Mobile IP architecture;

Figure 3 shows an exemplary illustrative network protocol enveloping that may be used to meet the possible required security policies to allow network traffic to flow between a mobile node to a foreign agent through policy enforcement equipment (e.g. firewall) to the home agent and then to another communications end point;

Figure 4 shows an example mobility architecture in accordance with a presently preferred exemplary illustrative non-limiting embodiment of the present invention;

Figures 5 & 5A-5F show illustrative usage scenarios;

Figure 5G shows an exemplary client-server architecture;

Figure 6 shows an example simplified prior art operating system security architecture;

Figure 7 shows the example illustrative Figure 6 architecture modified to provide secure transparent illustrative mobility functionality;

Figure 8 shows an example illustrative run time linking sample;

Figure 9 shows an example illustrative client policy agent hooking; and

Figure 10 shows an example illustrative server architecture.

DETAILED DESCRIPTION OF EXEMPLARY NON-LIMITING EMBODIMENTS

Figure 4 shows an exemplary overall illustrative non-limiting mobility architecture. The example mobility architecture includes a mobility client (MC) and a mobility server (MS). The mobility client may be, for example, any sort of computing device such as a laptop, a palm top, a Pocket PC, a cellular telephone, a desktop computer, or any of a variety of other appliances having remote connectivity capabilities. In one exemplary embodiment, mobility client MC comprises a computing-capable platform that runs the Microsoft Windows 2000/XP operating system having security (for example, IPSec) functionality but other implementations are also possible. The system shown is scaleable and can accommodate any number of mobility clients and mobility servers.

In one exemplary embodiment, mobility client MC may be coupled to a network such as the Internet, a corporate LAN or WAN, an Intranet, or any other computer network. Such coupling can be wirelessly via a radio communications link such as for example a cellular telephone network or any other wireless radio or other communications link. In some embodiments, mobility client MC may be intermittently coupled to the network. The system shown is not, however, limited to wireless connectivity – wired connectivity can also be supported for example in the context of computing devices that are intermittently connected to a wired network. The wireless or other connectivity can be in the context of a local area network, a wide area network, or other network.

In the exemplary embodiment, mobility client MC communicates with the network using Internet Protocol (IP) or other suitable protocol over at least one of a plurality of possible network interfaces. In the illustrative embodiment, mobility server (MS) is also connected to the network over at least one of a plurality of possible network interfaces. The mobility server MS may communicate with one or more peers or other computing devices. The exemplary Figure 4 architecture allows mobility client MC to securely

communicate with the peer hosts via the communications link, the network and/or the mobility server MS.

In more detail, the Figure 4 mobility server maintains the state of each mobile device and handles the session management required to maintain continuous connections to network applications. When a mobile device becomes unreachable because it suspends, moves out of coverage or changes its "point of presence" address, the mobility server maintains the connection to the network host by acknowledging receipt of data and queuing requests.

The exemplary mobility server also manages network addresses for the mobile devices. Each device running on the mobile device has a virtual address on the network and a point of presence address. A standard protocol (e.g., DHCP) or static assignment determines the virtual address. While the point of presence address of a mobile device will change when the device moves from one subnet to another (the virtual address stays constant while the connections are active).

This illustrative arrangement works with standard transport protocols such as TCP/IP – intelligence on the mobile device and the mobility server assures that an application running on the mobile device remains in sync with its server.

The mobility server also provides centralized system management through console applications and exhaustive metrics. A system administrator can use these tools to configure and manage remote connections, troubleshoot problems, and conduct traffic studies.

The mobility server also, in the exemplary embodiment, manages the security of data that passes between it and the mobile devices on the public airways or on a wireline network. The server provides a firewall function by giving only authenticated devices access to the network. The mobility server can also certify and optionally encrypt all communications between the server and the mobile device. Tight integration with Active Directory or other directory/name service provides centralized user policy management for security.

The Figure 4 architecture can be applied in any or all of a large and varying number of situations including but not limited to the exemplary situations shown in Figure 5 (for brevity and clarity sake, example embodiments are described using a single network point of attachment but it will be appreciated and understood that the current invention is not to be limited to such scope and application):

- At example location number one shown in Figure 5 and see also Figure 5A, the mobility client (depicted as a laptop computer for purposes of illustration) is shown inside a corporate or other firewall, and is shown connected to a wireless LAN (WLAN) having an access point. In this example, a private wireless network is connected to a wireline network through the mobility server. All application traffic generated on or destined for the wireless network is secured, and no other network traffic is bridged or routed to the wireless network. Using standard firewall features found in the operating system, the system can be further configured to allow only mobility traffic to be processed by the mobility server on the wireless network. In this example, the mobility client is authenticated to the mobility server. Packets flow normally between the mobility client and the mobility server, and the communication channel between the mobility client and the mobility server is protected using the conventional IPSec security protocol.
- At example location number two shown in Figure 5, the mobility client has moved into a dead-spot and lost connectivity with the network. The mobility server maintains the mobility client's network applications sessions during this time. Had the mobile client been using Mobile IP instead of the exemplary embodiment herein, the client's sessions could have been dropped because Mobile IP does not offer session persistence.
- At example location number three, the mobility client has moved back into range of the corporate network on a different subnet. The mobility client acquires a new point-of-presence (POP) address on the new subnet, negotiates a new secure channel back to the mobility server using IPSec,

reauthenticates with the mobility server, and resumes the previously suspended network sessions without intervention from the user and without restarting the applications. This process is transparent to the mobile applications and to the application server.

- At example location numbers four and five shown in Figure 5 and see also Figures 5B & 5C, the mobility client has left the corporate network and roamed into range of public networks. For example, the mobile client at location 4 shown in Figure 5 is shown in range of a conventional Wireless Wide Area Network (WWAN) wireless tower, and the mobile client at location 5 shown in Figure 5 is shown in range of a Wi-FI or other wireless access point "hot spot" such as found in an airport terminal, conference center, coffee house, etc. The wireless technology used for the public network need not be the same as that used inside the enterprise – since the illustrative system provides for secure roaming across heterogeneous networks. The mobility client's traffic must now pass through a corporate or other firewall. The firewall can be configured to pass IPSec traffic intended for the mobility server and/or the mobility client can be configured to use an IPSec session to the firewall. Either solution can be implemented without end-user interaction, although intervention is possible.
- In the Figure 5B example, mobility devices are connected to a diverse, public wide area network. The enterprise is also connected to the public network through a conventional firewall. The firewall is, in the exemplary embodiment, modified to allow mobility connections, specifically to the address of the mobility server. The connections are then protected by conventional security protocols such as IPSec.
- In the Figure 5C example, a private, wired network on a corporate, hospital, or other campus, and a wireless local area network supporting mobile devices connected to it through a conventional firewall. Traffic from the public to the private network that is not destined for the correct port is denied using conventional firewall rules. The firewall rules can specify either the domain ("allow access to 123.111.x:5008") or the addresses of

particular mobility servers ("allow access to 123.111.22.3:1002 and 123.111.23.4:5008") – the latter approach being more secure. On a smaller campus, a single, multi-homed mobility server could be used to handle both the wired and wireless LAN traffic. Once a user is authenticated, he or she has access to the wired network. A Network Address Translator (NAT) maybe used to reduce the number of public (routable) IP addresses required. In the example shown in Figure 5C, a many-to-one relationship is provided so that mobile devices can use just one of two IP addresses instead of requiring one address each. Any traffic coming from the wireless LAN access points preferably must satisfy both the firewall rules and be cleared by the mobility server. With encryption enabled, this configuration protects the wired network while offering legitimate wireless users full, secure access to corporate data.

- Figure 5D shows an example configuration that allows users to roam securely across different networks both inside and outside of the corporate firewall. The mobility server sits behind the firewall. When the mobility client is inside the corporate firewall, connected to the wireless LAN (WLAN), and has been authenticated to the mobility server, packets flow normally and the communication channel between the mobile device and the mobility server (mobile VPN) is protected using IPSec. In this example, the Public Key Infrastructure, passwords and/or any other desired mechanism can be used to perform the key exchange for the IPSec tunnel. For added protection, WLAN access points inside the firewall can be configured to filter all protocols except for a desired one (e.g., IPSec). The mobility server acts as a VPN protecting the data as it traverses the wireless network with IPSec encryption. In this exemplary configuration, the mobility server also acts as a firewall by preventing intruders from accessing the private network. When the mobile device (client) moves into range of the corporate network on a different subnet, it acquires a new point-of-presence (POP) address on the new subnet, negotiates a new secure channel back to the mobility server using IPSec, re-authenticates with the

mobility server, and resumes the previously suspended application sessions – all without user intervention being required. The applications can continue to run and the TCP or other connections can be maintained during this network transition since the network transition is transparent to the applications and the mobility server proxies communications on behalf of the mobile device during times when it is unreachable.

- The Figure 5E illustrative network configuration extends the protection of an enterprise firewall to its mobile clients. In this illustrative scenario, the mobility client is configured to use a conventional L2TP/IPSec tunnel to the firewall. IPSec filters on the mobile client can be configured to pass only authenticated IPSec packets to the mobile client's transport protocol stack and reject all other packets. The corporate firewall can be configured to reject all packets except for authenticated IPSec packets for trusted clients; any control channels necessary to set up secure connections; and responses to packets that originate from within the firewall for specifically permitted Internet or other network services. The mobility server located behind the firewall acts as a transport-level, proxy firewall. By proxying all network traffic, user transactions are forced through controlled software that protects the user's device from a wide variety of attacks including for example those using malformed packets, buffer overflows, fragmentation errors, and port scanning. Because the mobility server acts as a transport-level proxy, it can provide this protection transparently for a wide range of applications. Attacks against the network can be blocked by filter rules configured on the firewall and/or the proxy firewall capabilities of the mobility server. Attacks against the mobile device are prevented by the IPSec filter rules configured on the mobile client. Attempts to crack user passwords using sniffer attacks are thwarted by the secure tunnel provided by IPSec.
- Figure 5F shows an additional exemplary illustrative e-commerce model. Like WAP, the Figure 5F arrangement provides optimizations that enhance performance and reliability on slow and unreliable wireless networks. Unlike WAP, the Figure 5F system doesn't allow data to sit on an

intermediate server in an unencrypted state. The Figure 5F architecture allows standard web protocols such as HTTP and TLS to be used for e-commerce or other transactions (the web traffic is treated as a payload). The encrypted data is forwarded to its final destination (e.g., the web server) where is can be processed in the same way it would be if two wired peers were performing the same transaction. In addition to optimizations for wireless networks, the Figure 5F system provides seamless roaming between different networks and application session persistence while devices are suspended or out of range of a wireless base station. When combined with the illustrative system's support for public key infrastructure and/or other security mechanisms, those capabilities form a powerful mobile e-commerce platform.

The scenarios described above are only illustrative – any number of other intermittent, mobile, nomadic or other connectivity scenarios could also be provided.

Exemplary Integration with IPSec Standards-Based Security Framework

Generally, the IPSec process of protecting frames can be broadly handled by three logically distinct functions. They are:

- Policy configuration and administration
- Security negotiation/key management
- Privacy processing

Although these processes are logically distinct, the responsibility for implementing the functionality may be shared by one or more modules or distributed in any manner within an operating or other system. For instance, in the exemplary illustrative client operating system embodiment, the implementation is broken into 3 functional areas or logical modules:

1. A Policy Agent module
2. A security negotiation and key management (e.g., ISAKMP/IKE) module
3. A privacy (e.g., IPSec) module.

In this illustrative example, the Policy Agent is responsible for the configuration and storage of the configured policy -- however it is the IPSec module that actually acts upon the requested policy of the Policy Agent. The preferred exemplary illustrative system provides two different related but separated aspects:

- the first aspect handles IPSec from the mobility client to the firewall or the mobility server; and
- the other aspect handles communication on virtual addresses between the mobility server and peer hosts.

We first discuss exemplary illustrative communication to and from the mobility client.

EXAMPLE MOBILITY CLIENT ARCHITECTURE

As part of the preferred embodiment's overall design, network roaming activity is normally hidden from the applications running on the mobility client – and thus, the application generally does not get informed of (or even need to know about) the details concerning mobility roaming. Briefly, as described in the various copending commonly-assigned patent applications and publications referenced above, each of the mobile devices executes a mobility management software client that supplies the mobile device with the intelligence to intercept network activity and relay it (e.g., via a mobile RPC or other protocol) to mobility management server. In the preferred embodiment, the mobility management client generally works transparently with operating system features present on the mobile device to keep client-site application sessions active when contact is lost with the network. A new, mobile interceptor/redirector component is inserted at the conventional transport protocol interface of the mobile device software architecture. While mobile interceptor/redirector could operate at a different level than the transport interface, there are advantages in having the mobile interceptor/redirector operate above the transport layer itself. This mobile interceptor or redirector transparently intercepts certain calls at this interface and routes them (e.g., via

RPC and Internet Mobility Protocols and the standard transport protocols) to the mobility management server over the data communications network. The mobile interceptor/redirector thus can, for example, intercept network activity and relay it to server. The interceptor/redirector works transparently with operating system features to allow application sessions to remain active when the mobile device loses contact with the network.

This arrangement provides an advantageous degree of transparency to the application, to the network and to other network sources/destinations. However, we have found that IPSec is a special case. Between the mobility client and the mobility server or the mobility client and a firewall, IPSec is protecting the packets using the point-of-presence (POP) address. Therefore, in one exemplary embodiment, to allow the existing IPSec infrastructure to operate normally, it should preferably remain informed of the current state of the network. We have therefore modified our previous design to inform IPSec of the change of network status (e.g., so it can negotiate a IPSec session when network connectivity is reestablished) while continuing to shield the networked application and the rest of the operating system from the temporary loss of a network access,. Before describing how that is done in one illustrative embodiment, we first explain – for purposes of illustration only -- the conventional Microsoft Windows 2000/XP operating system IPSec architecture shown in Figure 6. Note that Windows 2000/XP and IPSec is described only for purposes of illustration – other operating systems and security arrangements could be used instead.

In Windows 2000/XP, the IPSec module is responsible for filtering and protecting frames. For additional information, see for example Weber, Chris, "Using IPSec in Windows 2000 and XP" (Security Focus 12/5/01). Briefly, however, by way of non-limiting illustrative example, before allowing a frame to be processed by the protocol stack or before transmitting the frame out on the network, the network stack first allows the IPSec module a chance to process the frame. The IPSec module applies whatever polices to the frame the Policy Agent requests for the corresponding network identity. In the

event that the Policy Agent requires the IPSec module to protect a frame but it does not yet have the required security association (SA) with the peer in accordance with the requested policy, it issues a request to the security negotiation/key management module -- in this illustrative case the ISAKMP/IKE (Internet Security Association and Key Management Protocol/Internet Key Exchange) module -- to establish one. It is the responsibility of the ISAKMP/IKE module in this illustrative system to negotiate the requested security association and alert the IPSec (privacy) module as to the progress/status of the security association. Once the security association has been successfully established, the IPSec module continues its processing of the original frame.

In the illustrative embodiment, the Policy Agent uses conventional Microsoft Winsock API's (application programming interfaces) to monitor the state of the network and adjust its policies accordingly. However this is implementation-dependent as other interfaces may also be used to alert this logical component of the network state in other environments. Accordingly, the ISAKMP/IKE module also uses conventional Microsoft Winsock API's to perform security association negotiation as well as track network state changes in one exemplary embodiment.

Briefly, the above techniques establish a secure IPSec session that is generally tied to a particular IP address and/or port and must be essentially continuous in order to be maintained, as is well known. If the secure session is temporarily interrupted (e.g., because of a lost or suspended connection or a roam) and/or if the IP address and/or port changes, IPSec will terminate it. Unless something is done, terminating the secure IPSec session will cause the mobile application to lose communication even if the network session continues to appear to remain in place. The preferred illustrative exemplary embodiment solves this problem by introducing functionality ensuring that IPSec is passed sufficient information to allow it to react to the secure session being lost while continuing to shield this fact from the application – and by allowing IPSec to (re)negotiate a secure session once the

network connectivity is reestablished using the same or different IP address or port number – all transparently to the networked application. In this way, the exemplary illustrative application is not adversely affected by termination of a previous security session and the establishment of a new one -- just as the application is not adversely affected by access to the previous network being terminated and then reestablished (or in the case of roaming, to a new network with a new network identity being provisioned in its place). Meanwhile, the mobility server during such interruptions continues to proxy communications with the peer(s) the mobile device is communicating with so that network application sessions are maintained and can pick up where they left off before the interruption occurred.

Mobility client-side and server-side support each have different requirements. Therefore the architectures are different in the exemplary illustrative embodiment. The block diagram of an exemplary client architecture is show in Figures 5G and 7. Note that as compared to conventional Figure 6, we have added two additional components:

- a Policy Agent Hooking component (nmplagnt), and
- a network virtualizing component (nmdrv).

Briefly, in the preferred illustrative embodiment, the network-virtualizing component virtualizes the underlying client module network while selectively allowing the core operating system's IPSec infrastructure to continue to be informed about network state changes. In the illustrative embodiment, the Policy Agent Hooking component "hooks" certain Policy Agent functions and redirects such processing to the network-virtualizing component so that the normal function of IPSec can be somewhat modified.

In more detail, in the exemplary embodiment, the network-virtualizing component (nmdrv) uses the services of the existing networking stack and is the layer responsible for virtualizing the underlying client module network. It also initiates and maintains the connection with the mobility server. When a client network application asks for the list of local IP addresses, the network-virtualizing component (nmdrv) intercepts the request and returns at

least one of a possible plurality of the mobility client's virtual network identities (e.g. virtual IP addresses).

However, to continue to allow the inherent IPSec components to operate in a normal fashion, the client architecture should preferably allow the associated IPSec modules to see and track the current point of presence (POP) network address(es). Therefore, in the exemplary embodiment, if a request for the list of network addresses is issued and the request originated in the IPSec process, the network-virtualizing module passes the request along to an inherent network stack without any filtering or modification. Therefore, both the Policy Agent (e.g., polagent.dll in Windows 2000, ipsecsvc.dll in Windows XP) and the ISAKMP/IKE module are kept abreast of the mobility client's current POP address(es).

In the exemplary embodiment, the network-virtualizing module also tracks address changes. Without this component, the network stack would normally inform any associated applications of address list changes through the conventional application-programming interface, possibly by terminating the application communications end point. In the Microsoft operating systems, for example, this responsibility is normally funneled through the conventional Winsock module, which in turn would then inform any interested network applications of the respective changes. In the exemplary embodiment, the Policy Agent registers interest with Winsock (e.g., using the SIO_ADDRESS_LIST_CHANGE IOCTL via the conventional WSAIoctl function) and waits for the associated completion of the request. The Policy Agent may also be event driven and receive asynchronous notification of such network state changes. Again, in the illustrative exemplary embodiment, the Policy Agent also registers with Winsock a notification event for signaling (e.g., on FD_ADDRESS_LIST_CHANGE via the WSAEventSelect function). When the Policy Agent is alerted to an address list change, it retrieves the current list of addresses, adjusts its policies accordingly and updates the associated policy administration logic. It further informs the Security Negotiation/Key Exchange module, in this case the ISAKMP/IKE module, of

the associated state change. The security negotiation/key exchange module (ISAKMP/IKE) module, in turn, updates its list of open connection endpoints for subsequent secure association (SA) negotiations.

In the exemplary embodiment, Winsock and associated applications are normally not allowed to see address list changes since this may disrupt normal application behavior and is handled by the network-virtualizing component. Therefore, in the preferred exemplary embodiment, another mechanism is used to inform the Policy Agent of changes with respect to the underlying network. To fulfill this requirement in the illustrative embodiment, the services of the Policy Agent Hooking module (nmplagnt) are employed.

To achieve the redirection of services, the illustrative embodiment employs the facilities of a hooking module (nmplagnt), and inserts the code into the policy administration, security negotiation, and key management (Policy Agent/ISAKMP/IKE) process(es) that are provided as part of the core operating system. In this illustrative embodiment, hooking only certain functions of the Policy Agent module to this redirected code is accomplished via a combination of manipulating the Import Address Table (IAT) together with the use of a technique known as code injection. Injection of the redirected functions is accomplished with the help of conventional operating system APIs (e.g. OpenProcess, VirtualAllocEx, ReadProcessMemory, WriteProcessMemory, and CreateRemoteThread) in the exemplary embodiment. In the preferred exemplary embodiment, once nmplagnt.dll is injected in lsass.exe executable module, it hooks LoadLibrary and FreeLibrary entries in lsasrv.dll so it can detect when the policy agent is loaded and unloaded. Of course, other implementations are possible depending on the particular operating environment.

Furthermore, the hooking technique in the illustrative embodiment takes advantage of the way in which the Microsoft Windows itself performs dynamic run-time linking. Generally, to facilitate code reuse, Microsoft Windows supports and uses extensively, Dynamic Link Libraries (DLLs). Through the use of DLL technology, a process is able to link to code

at run-time. To call a function in a dynamically linked library, the caller must know the location (address) of the specific function in the DLL. It is the operating systems responsibility to resolve the linkage between the code modules and is accomplished via an exchange of formatted tables present in both the caller and callee's run-time code modules. The dynamic library being called contains an Export Address Table (EAT). The Export Address Table contains the information necessary to find the specifically requested function(s) in the dynamic library. The module requesting the service has both an Import Lookup Table (ILT) and an Import Address Table (IAT). The Import Lookup Table contains information about which dynamic library are needed and which functions in each library are used. When the requesting module is loaded into memory for whatever reason, the core operating system scans the associated Import Lookup Table for any dynamic libraries the module depends on and loads those DLLs into memory. Once the specified modules are loaded, the requesting modules Import Address Table is updated by the operating system with the address(location) of each function that maybe accessed in each of the dynamically loaded libraries. Once again, in other environments, different implementations are possible.

In the exemplary embodiment, after the nmplagnt module is loaded by the prescribed method above, it hooks the Policy Agent's calls to the conventional Microsoft Windows Winsock functions WSASocket, WSAIoctl, WSAEventSelect, closesocket, and WSACleanup. After this process is executed, whenever the Policy Agent module attempts to register for notification of address changes, the request is redirected to the network-virtualizing component. As previously mentioned, the network-virtualizing component by design is aware of changes in network attachment. When it detects a change to the point of presence address, it sends the appropriate notifications to the Policy Agent module. In the illustrative embodiment, this causes the Policy Agent module to query for the current address list. Thus, the Policy Agent and consequently the ISAKMP/IKE module are informed of any address list changes.

Figure 8 is an example of how in the illustrative embodiment a single function from a single DLL might be linked into a calling process. In more detail, the operating system searched the ILT, found a need for target.dll, loaded target.dll into app.exe's address space, located TargetFunc in target.dll's EAT, and fixed up app.exe's IAT entry to point to TargetFunc in target.dll. Now when app.exe calls its stubbed TargetFunc, the stub function will call through the IAT to the imported TargetFunc. Because all of the calls of interest go through the IAT, the preferred exemplary embodiment is able to hook its target functions simply by replacing the corresponding entry for each function in the IAT as shown in Figure 9. This also has the advantage of localizing the hooking. Only the calls made by the requesting module in the target process are hooked. The rest of the system continues to function normally.

In summary, the illustrative embodiment in one exemplary detailed implementation performs the following steps:

1. Call OpenProcess to obtain access to the policy administration, security negotiation, and key management (Policy Agent/ISAKMP/IKE) process(es) and address space
2. Use ReadProcessMemory function to find the LoadLibrary function in the associated process(es)' address space.
3. Using the VirtualAllocEx function, allocate enough memory to hold the inject illustrative code shown in step 4 in the specified process(es)' address space.
4. Use WriteProcessMemory to inject the following code into the memory allocated in step 3:

```
LoadLibrary(&targetlibraryname);
label targetlibraryname:
   "C:\\Program Files\\NetMotion client\\nmplagnt.dll"
```

The address of the LoadLibrary function was determined in step 2. The data bytes at label targetlibraryname will vary depending on the name of

the module being loaded, where the corresponding module is located, and the operating system environment.

5. Call the CreateRemoteThread function to run the injected code.
6. Wait for the remote thread to exit.
7. Free the allocated memory
8. Close the process.

At the end of these steps, the nmplagnt module has been injected into the policy administration, security negotiation, and key management (Policy Agent/ISAKMP/IKE) process(es) where it is able to redirect the processing of the needed function calls. It is understood that the above code procedure is operating system and processor dependent and is only shown for illustrative purposes, thus not limiting to this specific sequence or operation. Furthermore, the executable code responsible for adding these components to the operating environment can be provided to the mobile device via storage on a storage medium (e.g., optical disk) and/or by downloading over the network

A similar method is employed using the FreeLibrary function instead of LoadLibrary function to reverse the hooking process and to unload the nmplagnt module. For the sake of brevity, the description is kept minimal, as anyone schooled in the art should be able to achieve the desired results.

EXAMPLE MOBILITY SERVER ARCHITECTURE

Using IPSec methodology for communication between the mobility server and peer hosts is a different set of problems to solve -- although it uses some of the same techniques used on the mobility client. Figure 10 shows an exemplary server architecture. In the exemplary illustrative embodiment, the mobility server MS can also be based on a Windows 2000/XP (or any other) operating system. In this particular illustrative implementation, a

hooking module is also used in the illustrative embodiment – but the functions intercepted by the hooking module in the case of mobility server MS are redirected to the proxy and filter modules that are also supplied by the preferred exemplary embodiment, instead of the network-virtualization module.

In the exemplary mobility server, a proxy driver (nmproxy) can be used to implement the bulk of the mobility server functionality. However, in one exemplary implementation, there are three separate problems to solve for which three additional logical modules are used. They are:

- a network identity mapping driver (nmprtmap),
- an IPSec filter driver (nmipsec), and
- the security negotiation hooking library (nmike).

The first problem is how to manage virtual addresses for the mobility clients. Although it is possible in some network stack implementations to assign multiple addresses to the inherent networking stack components of the operating system, some systems do not support such functionality. To support the more restrictive implementation, the illustrative example embodiment employs the use of an identity mapping technique. It will be appreciated that the techniques herein are both compatible with and complimentary to either implementation, and such identity mapping functionality allows the security functionality to successfully operate within the more restrictive environments. The illustrative mobility server opens a communications endpoint associated with a local address and port and then identity maps between the corresponding virtual address(es) and port(s) before packets are processed by the protocol stack during reception and before they are transmitted out on the network. That mapping is the job of the network identity mapping module (nmprtmap) in one exemplary embodiment. For example, assume an application on a mobility client opens TCP port 21 on virtual address 10.1.1.2. Through the use of previously-defined mechanisms (see for example U.S. Patent Application Serial No. 09/330,310 filed June 11,

1999, entitled "Method And Apparatus For Providing Mobile and Other Intermittent Connectivity In A Computing Environment), this request is transferred from the mobility client to the mobility server. In response, the MS opens the connection on its local address 10.1.1.1 on port 2042 and registers the appropriate mapping with the network identity mapping module. When the mobile client together with proxy driver (nmproxy) then wishes to send data on its newly opened connection, the packet generated by the inherent networking stack will have a source address of 10.1.1.1 port 2042. The network identity mapping module will then match the frame's protocol/address/port tuple against its mapping table and replace the source address with 10.1.1.2 port 21 before the packet is transmitted on network. The reverse operation is performed for received packets. Using this network identity mapping technique allows the mobility server to communicate to peer systems using virtualized addresses without requiring modification to the core operating system transport protocol stack

The second problem is a direct result of this mapping technique. Because the network identity mapping module logically operates below IPSec module (i.e. processes frames before during reception and after during transmission), it cannot directly manipulate IPSec protected frames without corrupting the packets or being intimately involved in the privacy or authenticating process. To address this issue, in one exemplary embodiment, the aforementioned IPSec filter module (nmipsec) inserts itself between the operating systems networking stack components and the associated IPSec modules. The filter module inspects each outgoing packet before IPSec protects the packet and each incoming packet after IPSec removes any encoding. Once in control of the frame, it consults the network identity mapping module(nmprtmap) to determine whether or not the frames source or destination identity should be mapped. In this way, the functionality of the mapping logic is moved to a level where is can perform its function without interfering with the IPSec processing.

Hooking the link between the IPSec and networking stack components is implementation and operating system dependent. In the illustrative exemplary embodiment, again the hooking process is completed by the manipulation of tables that are exchanged between the inherent IPSec and networking stack modules – but other implementations and environments could rely on other techniques. In the illustrative embodiment the IPSec filter module (nmipsec) loads before the IPSec module but after the transportprotocol module. When the IPSec module attempts to exchange its function table with the transport protocol components, the IPSec filter module (nmipsec) records and replaces the original function pointers with it's own entry points. Once the associated tables are exchanged in this manner, the IPSec filter module (nmipsec) can manipulate the contents of and control which packets the inherent IPSec module operates on.

The third issue is where the hooking techniques also used by the mobility clients is employed. As mentioned previously, due to the mapping technique employed in one exemplary implementation, the inherent networking stack has no knowledge of the mobility client's virtual address(es). Consequently, the policy administration, security negotiation, and key management (Policy Agent/ISAKMP/IKE module) process(es) are also not cognizant of these additional known network addresses. Therefore, there are no IPSec security policies to cover frames received for or transmitted from the mobility client's virtual address(es). Furthermore the security negotiation module (in this case the ISAKMP/IKE module) has no communications end point opened for which to negotiate security associations for the mobility client. To address this issue in the exemplary embodiment, the security negotiation hooking module(nmike) can employ the same hooking methodology described for the mobility client and illustrated in Figure 8. The security negotiation hooking module (nmike) intercepts any address change notification request. When the proxy modules registers or deregisters a mobility client's virtual address(es) with the network identity mapping module(nmprtmap), it also informs the security negotiation hooking

module(nmike). This module in turn then informs the policy administration module (Policy Agent) of the respective change. When either the policy administration (Policy Agent) or the security negotiation (ISAKMP/IKE) module requests a list of the current addresses via the conventional Microsoft Windows GetIpAddrTable function call, the security negotiation hooking module(nmike) intercepts the request and adds all of the current virtual addresses to the returned list. When the policy administration module (Policy Agent) sees the respective virtual addresses in the list, it treats them as actual addresses and creates the appropriate policies for the IPSec module. In response to the modification of the network address list, the security negotiation (ISAKMP/IKE) module will attempt to open and associate a communications endpoint for each address in the list. However, as mentioned previously, since the inherent networking stack in the illustrative embodiment is ignorant to the fact of these additional network addresses due to the aforementioned mapping methodology, this operation will generally fail. To solve this problem, the security negotiation hooking module (nmike) intercepts the request to the conventional Microsoft windows Winsock bind function and modifies the requested virtual address and port with a INADDR_ANY. Once the endpoint is bound through the inherent transport protocol stack, the security negotiation hooking module (nmike) employs the services of the network identity mapping module (nmprtmap) and creates a mapping between the actual address and port associated with the newly established communications end point to the virtual address and the assigned port for security negotiations (in this case port 500 is the standard ISAKMP port). Finally, the security negotiation hooking module (nmike) registers the actual address and port with IPSec filtering module (nmipsec) to instruct the module to pass packets to and from the specified address without further IPSec filter processing.

All documents referenced herein are incorporated by reference as if expressly set forth herein.

While the invention has been described in connection with practical and preferred embodiments, it is not to be so limited. Specifically, for example, the invention is not limited to IPSec or Microsoft operating systems. IPSec and related technologies can be arranged in a number of manners, executing with some of the required algorithms executing either in software or hardware. To wit, certain implementations may include hardware accelerator technology for the ciphering process, etc. Many network interface and computer manufactures have commercially available products that are used for this exact purpose. It is to be appreciated that the above specifications however describes the logical placement of required functionally and may actually execute in a distributed fashion. Accordingly, the invention is intended to cover all modifications and equivalent arrangements within the scope of the claims.

WE CLAIM:

1. A method of maintaining network communications with a mobile or other intermittently connected computing device executing at least one networked application that participates in at least one network application session, comprising:

(a) detecting the occurrence of an event affecting network communications with the computing device, and

(b) in response to said detection, terminating, instantiating, and/or reinstantiating an IP Security session for use by said computing device while maintaining said network application session(s).

2. The method of claim 1 wherein said detecting comprises detecting a change in network point of attachment.

3. The method of claim 1 wherein said detecting comprises detecting that an interruption of network connectivity has caused a previous IP Security session to be terminated.

4. The method of claim 1 wherein said detecting comprises detecting that the mobile device's network identity has changed.

5. The method of claim 1 wherein said detecting comprises detecting that the mobile device has roamed to a different network or subnetwork.

6. The method of claim 1 wherein said step (b) comprises negotiating a new IP Security session to replace a previous, lost IP Security session in a manner that is transparent to the networked application.

7. The method of claim 1 wherein said step (b) includes using IPSec to create a secure tunnel through the network.

8. The method of claim 1 further including applying policy rules to selectively allow, deny or delay the flow of network communications over said IP Security session.

9. The method of claim 1 further including centrally managing and distributing policy regarding the establishment of said IP Security session from a central authority.

10. The method of claim 1 further including securely proxying said mobile device communications.

11. The method of claim 1 further including terminating a previous IP Security session based on said detecting.

12. A method of modifying an operating environment having at least one software component, said operating environment using transport engine protocols and running at least one application, the method comprising:
(a) transparently and selectively injecting computer instructions into said operating environment; and
(b) redirecting the execution path of said at least one software component to achieve additional functionality while maintaining binary compatibility with said operating environment component(s), said transport engine protocols and said applications.

13. The method of claim 12 wherein said redirecting is performed based on process name.

14. A method of providing data communications in a mobile computing environment, said environment including at least one device using at least one network interface for network applications and operating system components, comprising:

(a) selectively and transparently virtualizing said at least one network interface, thereby shielding said network applications and operating system components from at least some characteristics of said mobile computing environment, and
(b) allowing other said components to remain cognizant of at least interruptions in connectivity and changes in network point of attachment.

15. A method for providing data communications in an environment including at least one device using at least one network interface for network applications and operating system components, comprising:

(a) selectively virtualizing said at least one network interface, thereby shielding said network applications and operating system components from at least some adverse events that may otherwise disturb communications; and

(b) using said virtualized network interface to conduct data communications.

16. The method of claim 15 wherein said adverse events include changes in network point of attachment.

17. The method of claim 15 wherein said adverse events include periods of network disconnectedness.

18. A method for using plural IP Security sessions over a plurality of network interfaces associated with at least one network point of attachment, comprising:

(a) distributing network application communications to simultaneously flow over said plural IP Security sessions, and

(b) multiplexing/demultiplexing said distributed communication flows into corresponding higher layer communications sessions.

19. The method of claim 18 further including applying policy rules to selectively allow, deny, or delay the flow of network communications over at least one of said plural IP Security sessions.

20. The method of claim 18 further including centrally managing and distributing policy regarding the establishment of said plural IP Security sessions from a central authority.

21. A method comprising:

(a) facilitating the creation of plural IP Security sessions; and

(b) selectively allowing, denying and/or delaying the flow of network communications over at least one of said plural IP Security sessions based at least in part on applying policy rules.

22. The method of claim 21 further including centrally managing and distributing said policy rules from a central authority.

23. A method of administering secure network connections comprising:

(a) establishing IP Security sessions within a computing network; and

(b) centrally managing and distributing policy regarding the establishment of said IP Security sessions from a central authority.

24. A method of proxying mobile communications comprising:

(a) establishing communications with a mobile device;

(b) establishing communications with an ultimate peer of said mobile device; and

(b) instantiating at least one of a possible plurality of IP Security sessions with said ultimate peer on behalf of said mobile device.

25. The method of claim 24 wherein said mobile device includes a client and (a) comprises establishing client-server communications.

26. A method of proxying mobile communications comprising:

(a) establishing at least one IP Security session between said mobile device and a communication peer thereof; and

(b) maintaining said IP Security session with said communication peer during periods when said mobile device is unreachable.

27. A method of managing IP Security sessions between a mobility server and an ultimate communications peer, comprising:

(a) establishing at least one IP Security session between said mobility server and said ultimate communications peer; and

(b) automatically terminating said IP Security session in response to occurrence of a predetermined event.

28. The method of claim 27 wherein the predetermined event is selected from the group comprising link activity, application session inactivity, and termination of a communications end point.

29. A method of providing secure communications between a mobility client having a network identity, a mobility server and an ultimate communications peer, comprising:

(a) establishing at least one IP Security session between the mobility server and the ultimate peer; and

(b) securely maintaining said IP Security session even when the network identity of said mobility client changes.

30. In a system for maintaining network communications with a mobile or other intermittently connected computing device executing at least one

networked application that participates in at least one network application session, said system comprising:

a detector that detects the occurrence of an event affecting network communications with the computing device, and

a security module that, in response to said detection, instantiates or reinstantiates an IP Security session for use by said computing device while maintaining said network application session(s).

31. The system of claim 30 wherein said detector detects a change in network point of attachment.

32. The system of claim 30 wherein said detector detects that an interruption of network connectivity has caused a previous IP Security session to be terminated.

33. The system of claim 30 wherein said detector detects that the mobile device's network identity has changed.

34. The system of claim 30 wherein said detector detects that the mobile device has roamed to a different network or subnetwork.

35. The system of claim 30 wherein said security module negotiates a new IP Security session to replace a previous, lost IP Security session in a manner that is transparent to the networked application.

36. The system of claim 30 wherein said security module uses IPSec to create a secure session through the network communication.

37. The system of claim 30 further including a policy manager that applies policy rules to selectively allow, deny or delay the flow of network communications over said IP Security session.

38. The system of claim 30 further including a central policy management authority that centrally manages and distributes policy regarding the establishment of said IP Security session .

39. The system of claim 30 further including a mobility server that securely proxies said mobile device communications.

40. The system of claim 30 wherein the security module terminates a previous IP Security session based on said detection.

41. An operating environment having at least one software component, said operating environment using transport engine protocols and running at least one application, the environment further comprising computer instructions transparently and selectively injected therein, wherein the injected computer instructions include a redirector that redirects the execution path of said at least one software component to achieve additional functionality while maintaining binary compatibility with said operating environment component(s), said transport engine protocols and said applications.

42. The environment of claim 41 wherein said redirector redirects said execution path based on process name.

43. A mobile computing environment including at least one device using at least one network interface for network applications and operating system components, said environment comprising:
(a) instructions that selectively and transparently virtualize said at least one network interface, thereby shielding said network applications and operating system components from at least some characteristics of said mobile computing environment, and

(b) further instructions that allow other said components to remain cognizant of at least interruptions in connectivity and changes in network point of attachment.

44. An environment including at least one device using at least one network interface for network applications and operating system components, said environment comprising:

instructions that selectively virtualize said at least one network interface, thereby shielding said network applications and operating system components from at least some adverse events that may otherwise disturb communications; and

additional structure that uses said virtualized network interface to conduct data communications.

45. The environment of claim 44 wherein said adverse events include network point of attachment.

46. The environment of claim 44 wherein said adverse events include periods of network disconnectedness.

47. A system for using plural IP Security sessions over a plurality of network interfaces associated with at least one network point of attachment, comprising:

a data distributor that distributes network application communications to simultaneously flow over said plural IP Security sessions, and

(b) a multiplexer/demultiplexer that multiplexes and demultiplexes said distributed communication flows into corresponding higher layer communications sessions.

48. The system of claim 18 further including applying policy rules to selectively allow, deny, or delay the flow of network communications over at least one of said plural IP Security sessions.

49. The system of claim 47 further including a central authority that centrally manages and distributes policy regarding the establishment of said plural IP Security sessions.

50. A system comprising:

(a) a security framework that facilitates the creation of plural IP Security sessions; and

(b) a policy agent that selectively allows, denies and/or delays the flow of network communications over at least one of said plural IP Security sessions based at least in part on policy rules.

51. The system of claim 50 further including a central authority that centrally manages and distributes said policy rules.

52. A system for administering secure network connections comprising:

a security framework that establishes IP Security sessions within a computing network; and

a central authority that centrally manages and distributes policy regarding the establishment of said IP Security sessions.

53. A mobility proxy comprising:

a communications structure that establishes communications with a mobile device and with an ultimate peer of said mobile device; and

a security component that instantiates at least one of a possible plurality of IP Security sessions with said ultimate peer on behalf of said mobile device.

54. The system of claim 53 wherein said mobile device includes a client and mobility proxy comprises a server.

55. A system for proxying mobile communications comprising:

communications means for establishing at least one IP Security session with said mobile device and a communication peer thereof; and

a means for maintaining said IP Security session with said communication peer during periods when said mobile device is unreachable.

56. A system for managing IP Security sessions between a mobility server and an ultimate communications peer, comprising:

means for establishing at least one IP Security session between said mobility server and said ultimate communications peer; and

means for automatically terminating said IP Security session in response to occurrence of a predetermined event.

57. The system of claim 56 wherein the predetermined event is selected from the group comprising link activity, application session inactivity, and termination of a communications end point.

58. A system for providing secure communications between a mobility client having a network identity, a mobility server and an ultimate communications peer, comprising:

means for establishing at least one IP Security session between the mobility server and the ultimate peer and the mobility client and the mobility server; and

means for securely maintaining said IP Security session even when the network identity of said mobility client changes.

59. A storage medium storing:

a first set of instructions that inserts a policy agent hooking run-time linkable module into an operating system having a policy agent and an IPSec infrastructure, said hooking module informing the policy agent of network state changes; and

a second set of instructions that inserts a network interface virtualizing driver into said operating system, said virtualizing driver virtualizing a client module network and initiating mobility server connections while selectively allowing the IPSec infrastructure to continue to be informed about network state changes.

60. A method of preparing a mobile device for secure communications, said mobile device having an operating environment including a policy agent and an IPSec infrastructure, said method comprising:

downloading over a computer network onto the mobile device and executing with the mobile device, a first set of instructions that insert a policy agent hooking run-time linkable module into the operating environment, said hooking module informing the policy agent of network state changes; and

downloading over the computer network and executing with the mobile device a second set of instructions that inserts a network interface virtualizing driver into said operating environment, said virtualizing driver virtualizing a client module network and initiating mobility server connections while selectively allowing the IPSec infrastructure to continue to be informed about network state changes.

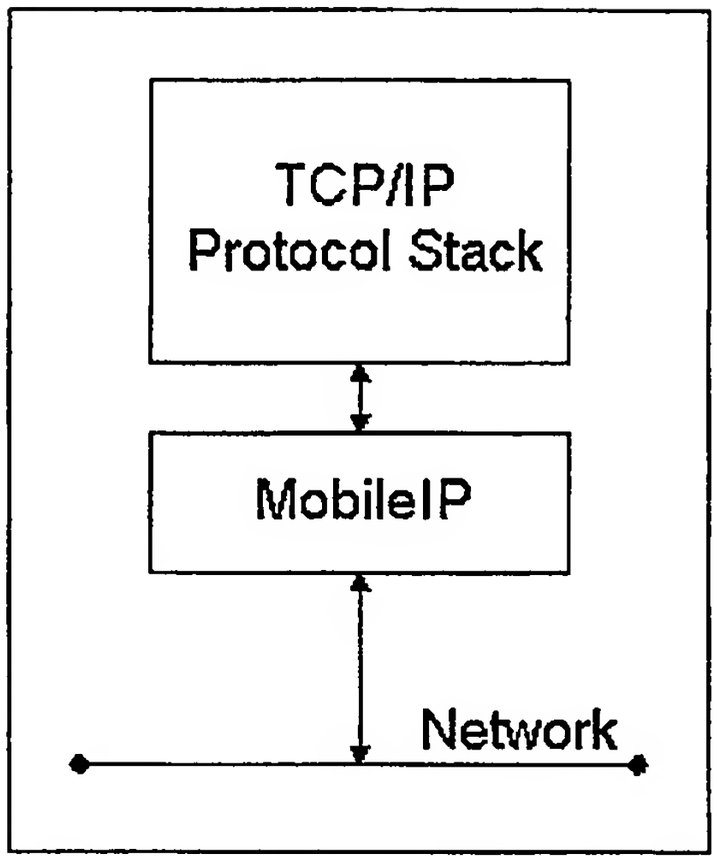


Figure 1 – Example Mobile IP Client Architecture (Prior Art)

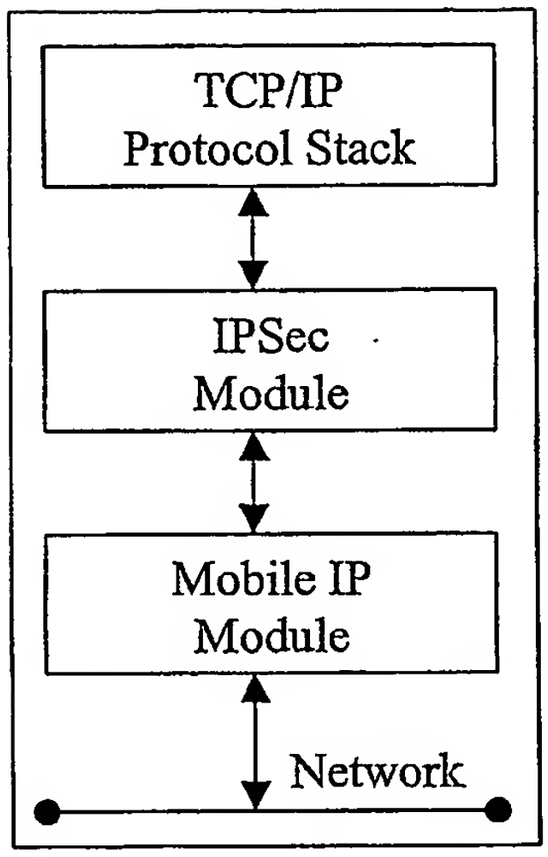


Figure 2: Example IPSec and Mobile IP Architecture (Prior Art)

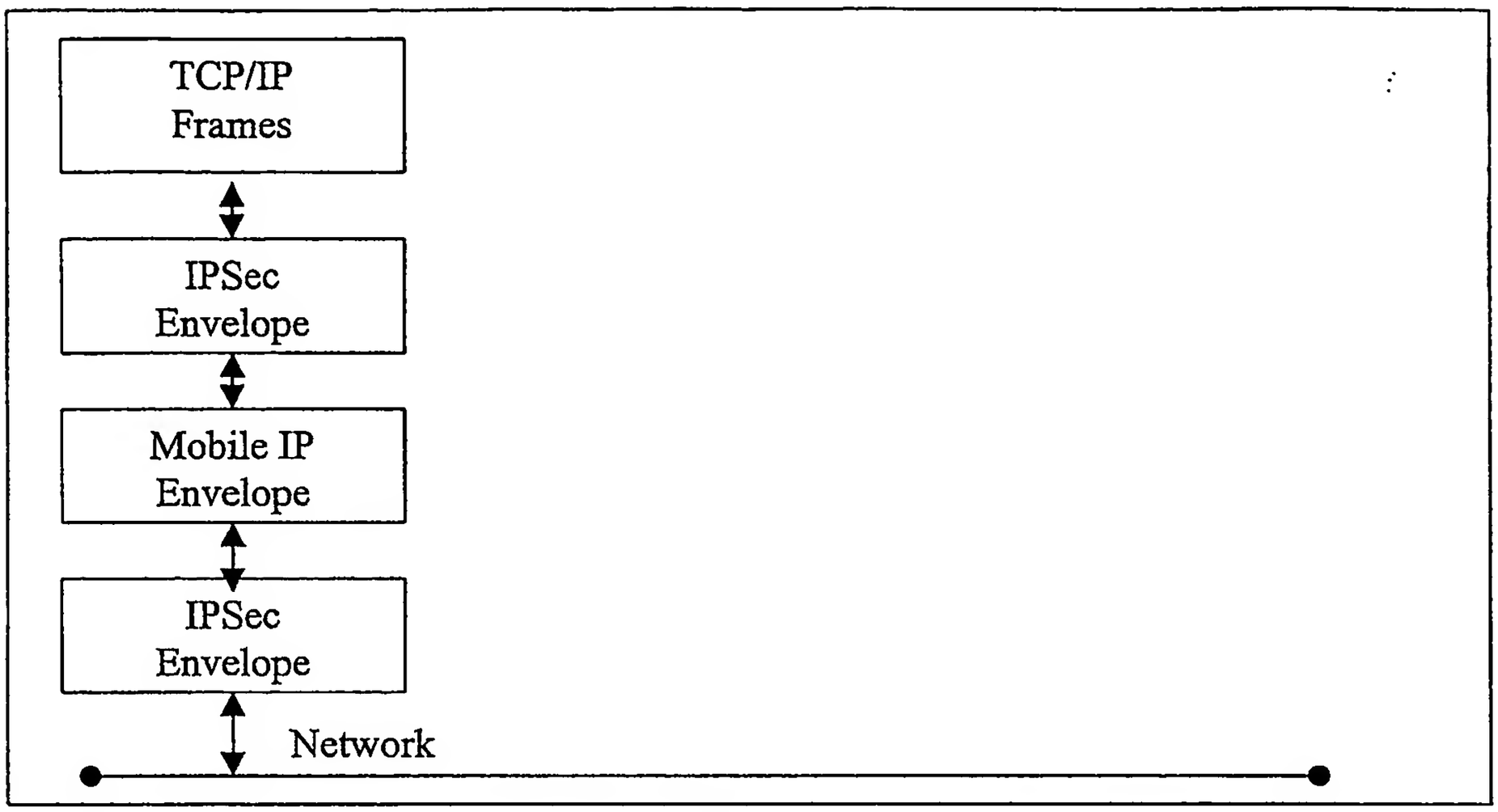


Figure 3: Enabling Roaming With Mobile IP And IPSec Encapsulation

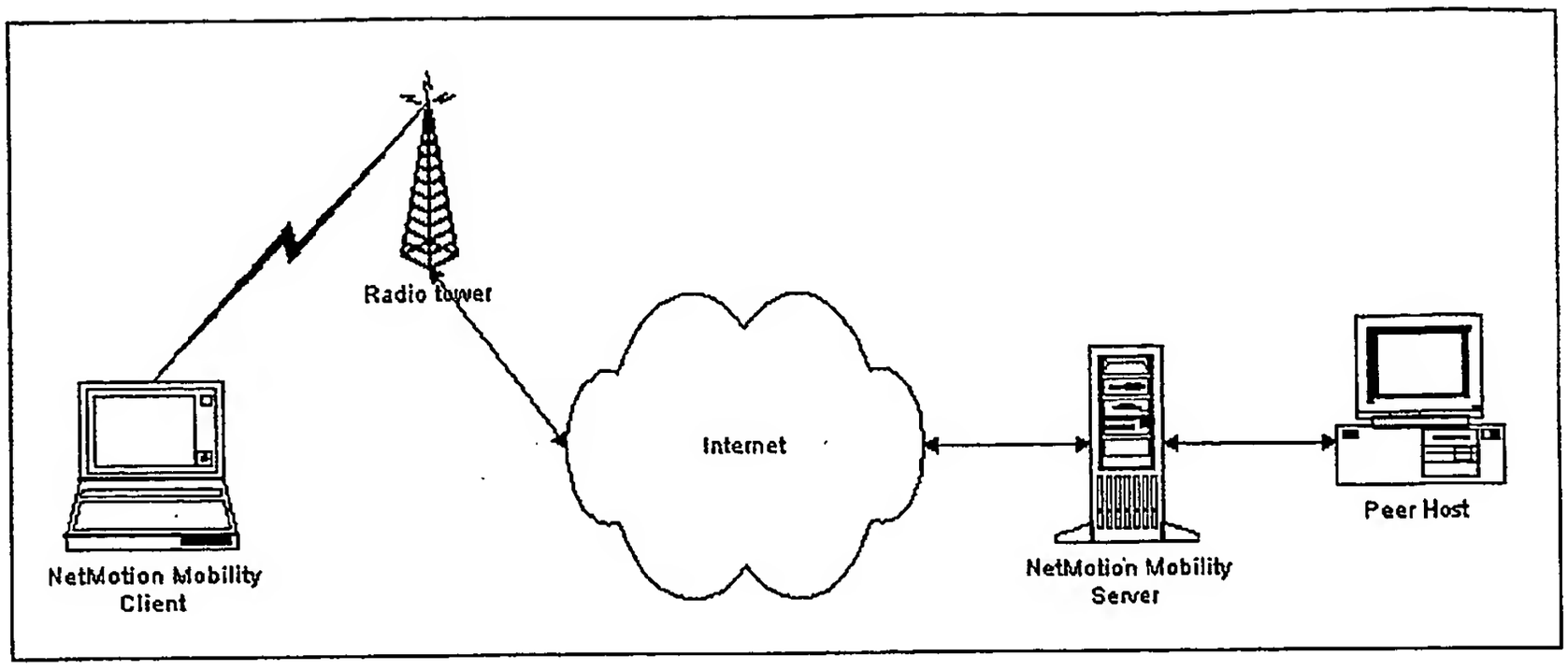


Figure 4 – Example Non-Limiting Secure Mobility Architecture

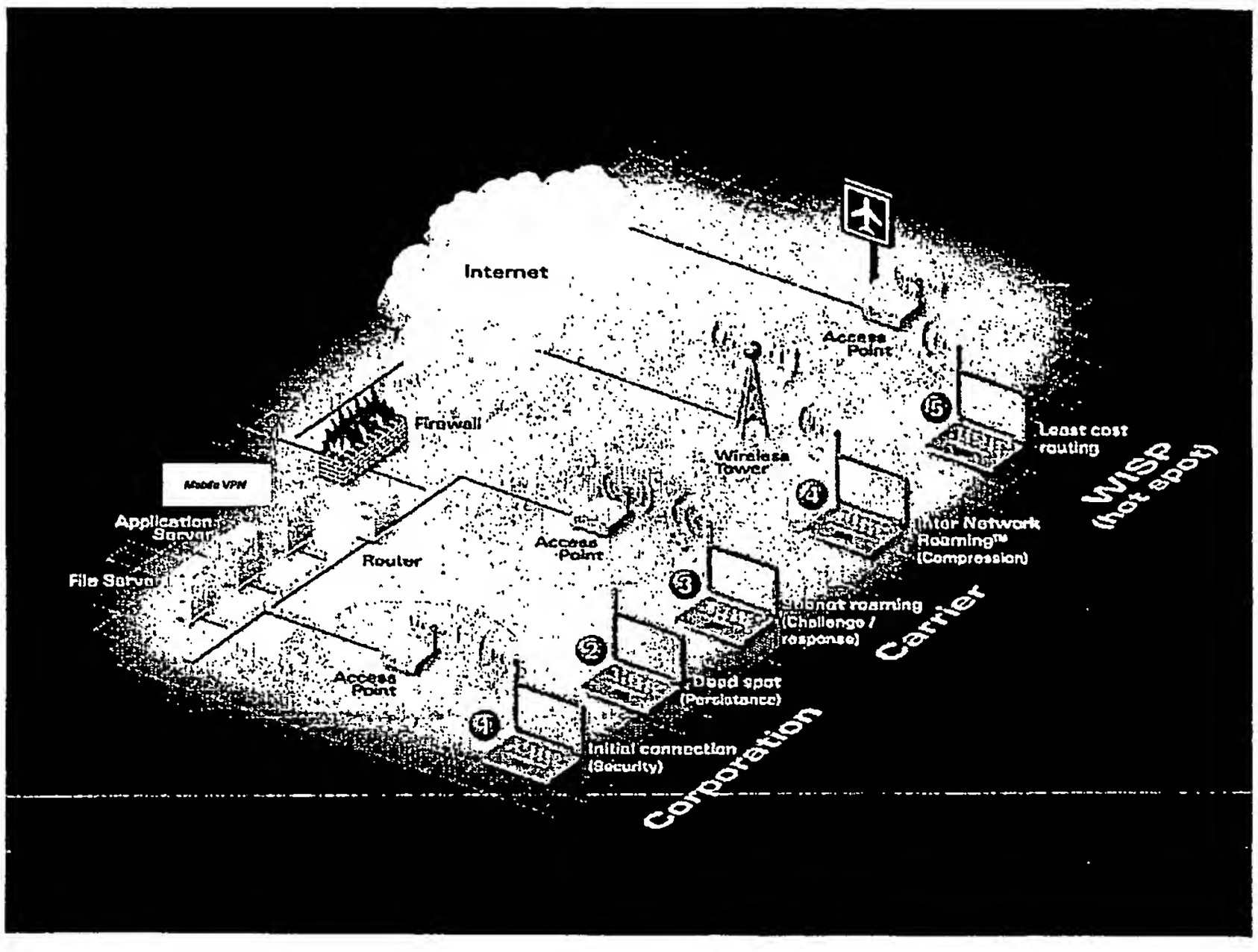


Figure 5 – Example Non-Limiting Illustrative Mobile Usage Scenarios

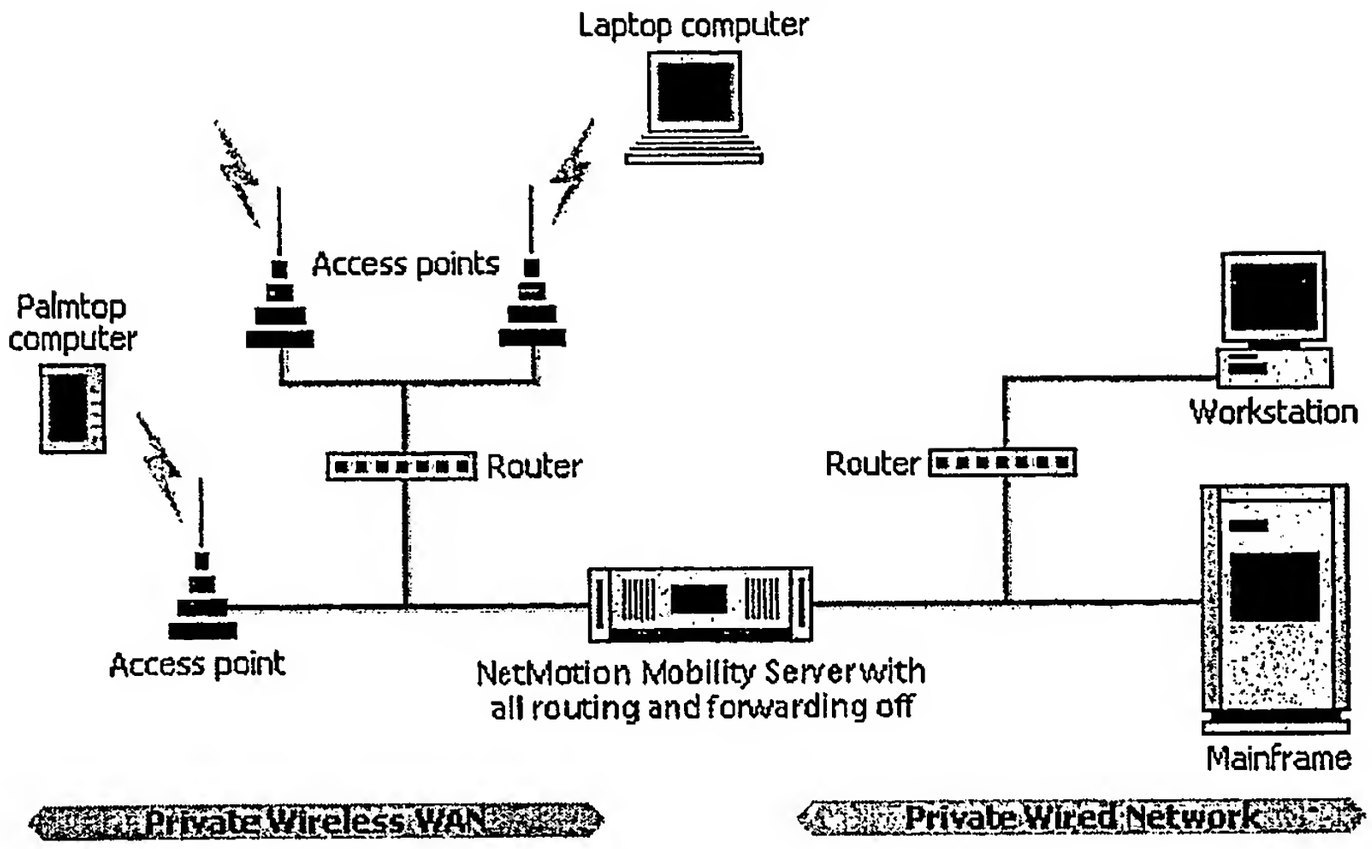

Laptop computer
Access points
Palmtop computer
Router
Router
Workstation
Access point
NetMotion Mobility Server with all routing and forwarding off
Mainframe
Private Wireless WAN
Private Wired Network

Figure 5A

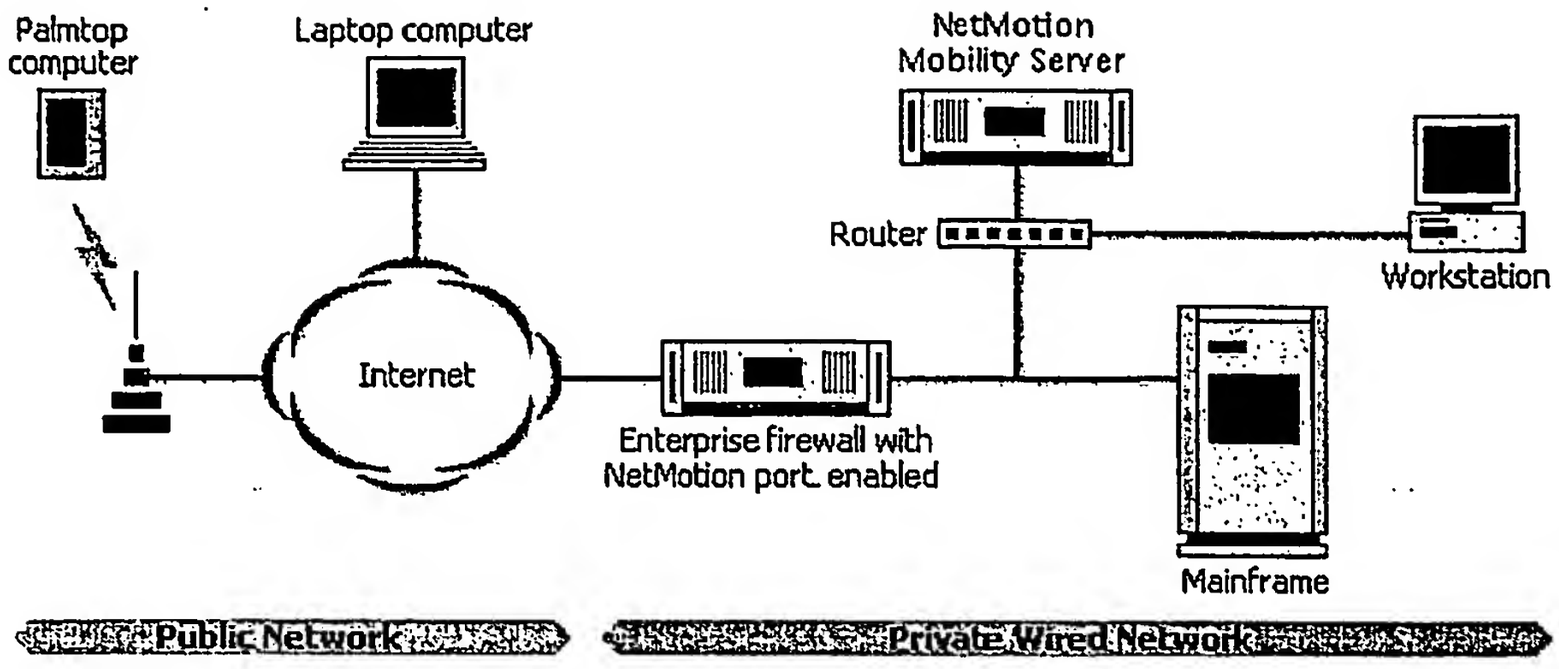

Palmtop computer
Laptop computer
NetMotion Mobility Server
Router
Workstation
Internet
Enterprise firewall with NetMotion port enabled
Mainframe
Public Network
Private Wired Network

Figure 5B

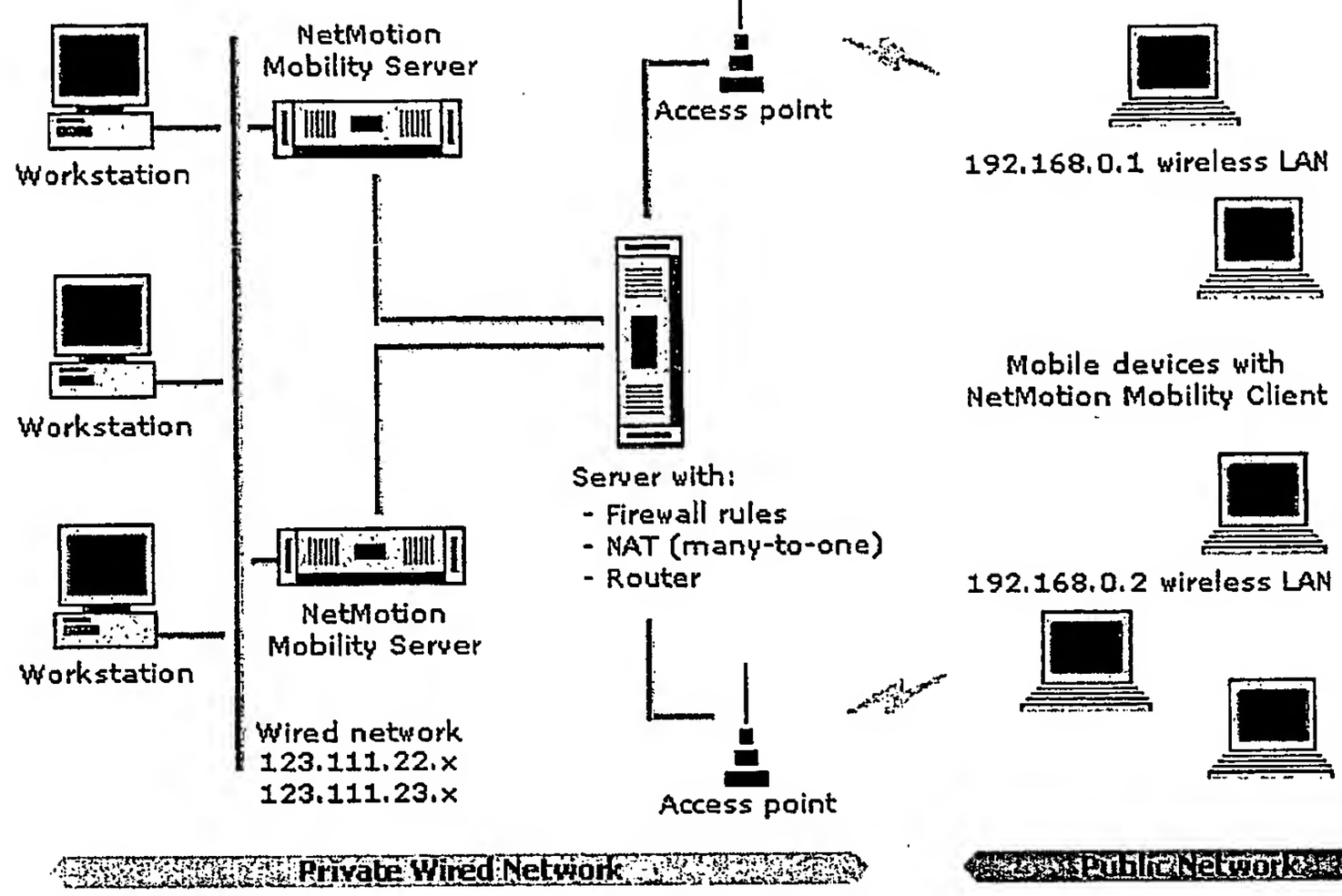

NetMotion
Mobility Server
Workstation
Workstation
Workstation
NetMotion
Mobility Server
Wired network
123.111.22.x
123.111.23.x
Access point
Server with:
- Firewall rules
- NAT (many-to-one)
- Router
Access point
192.168.0.1 wireless LAN
Mobile devices with
NetMotion Mobility Client
192.168.0.2 wireless LAN
Private Wired Network
Public Network

Figure 5C

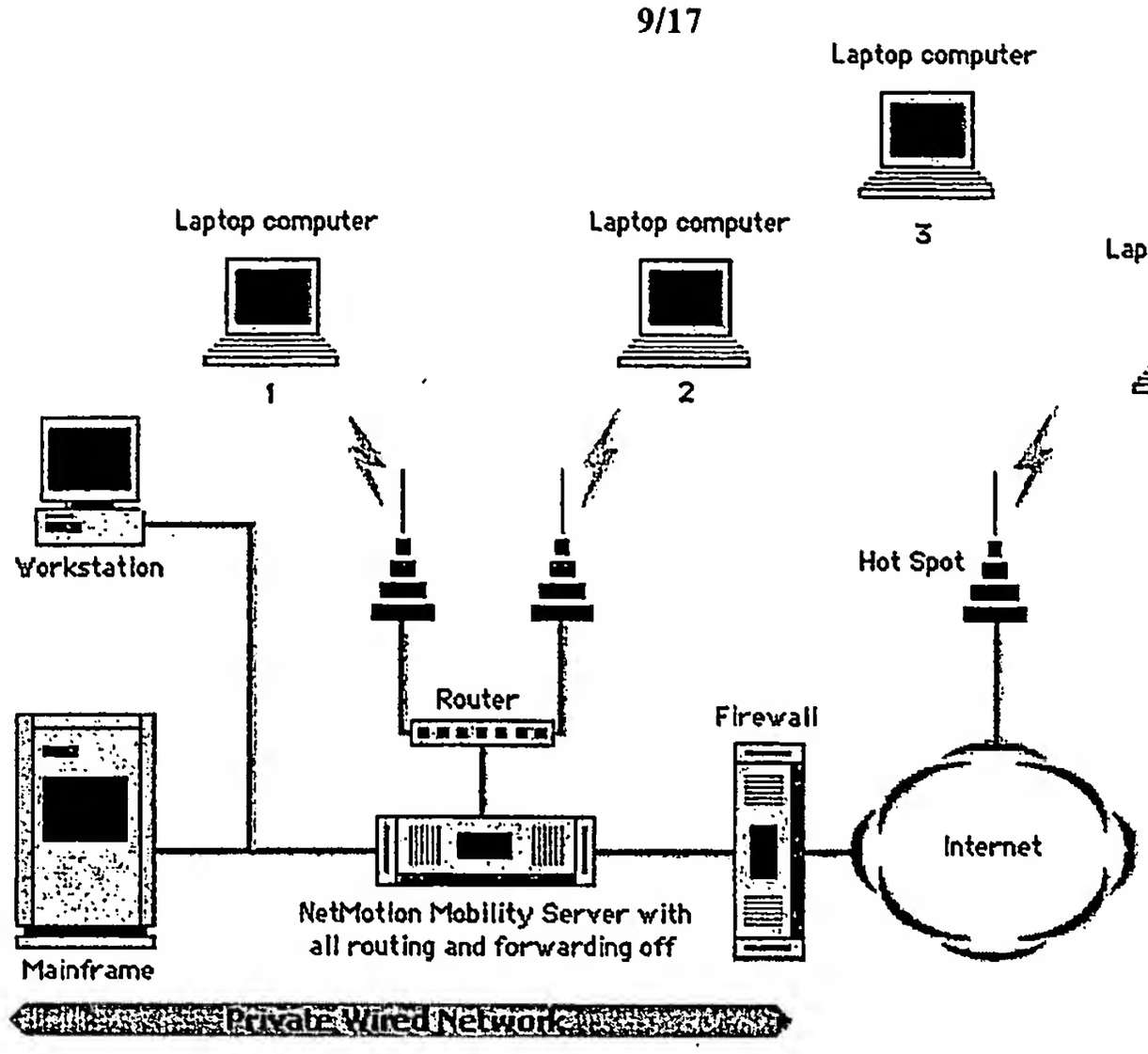

Laptop computer
1
Laptop computer
2
Laptop computer
3
Laptop computer
4
Workstation
Router
Hot Spot
Firewall
Internet
Mainframe
NetMotion Mobility Server with
all routing and forwarding off
Private Wired Network

Figure 5D

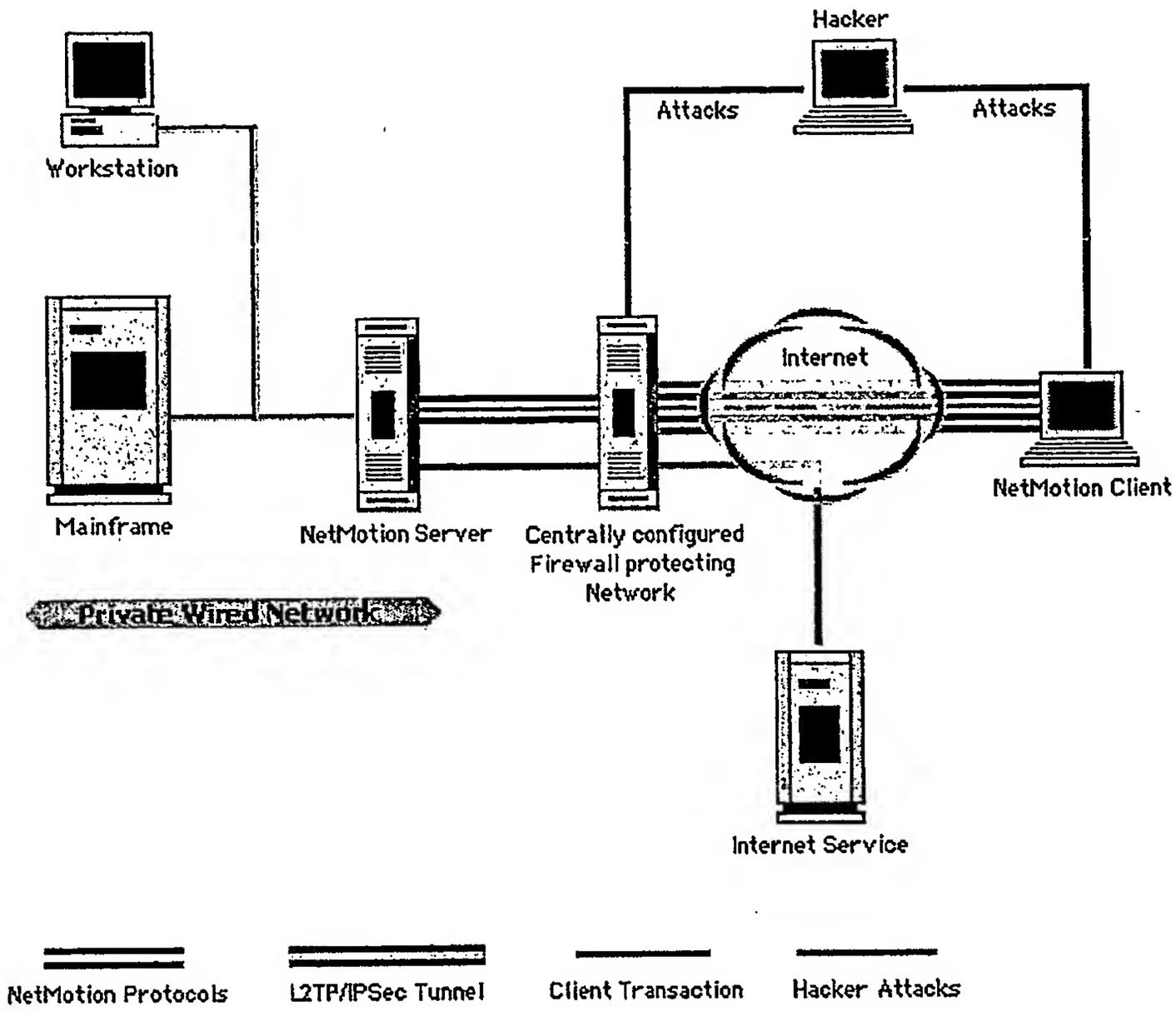

Hacker
Attacks
Attacks
Workstation
Internet
Mainframe
NetMotion Server
Centrally configured
Firewall protecting
Network
NetMotion Client
Private Wired Network
Internet Service
NetMotion Protocols
L2TP/IPSec Tunnel
Client Transaction
Hacker Attacks

Figure 5E

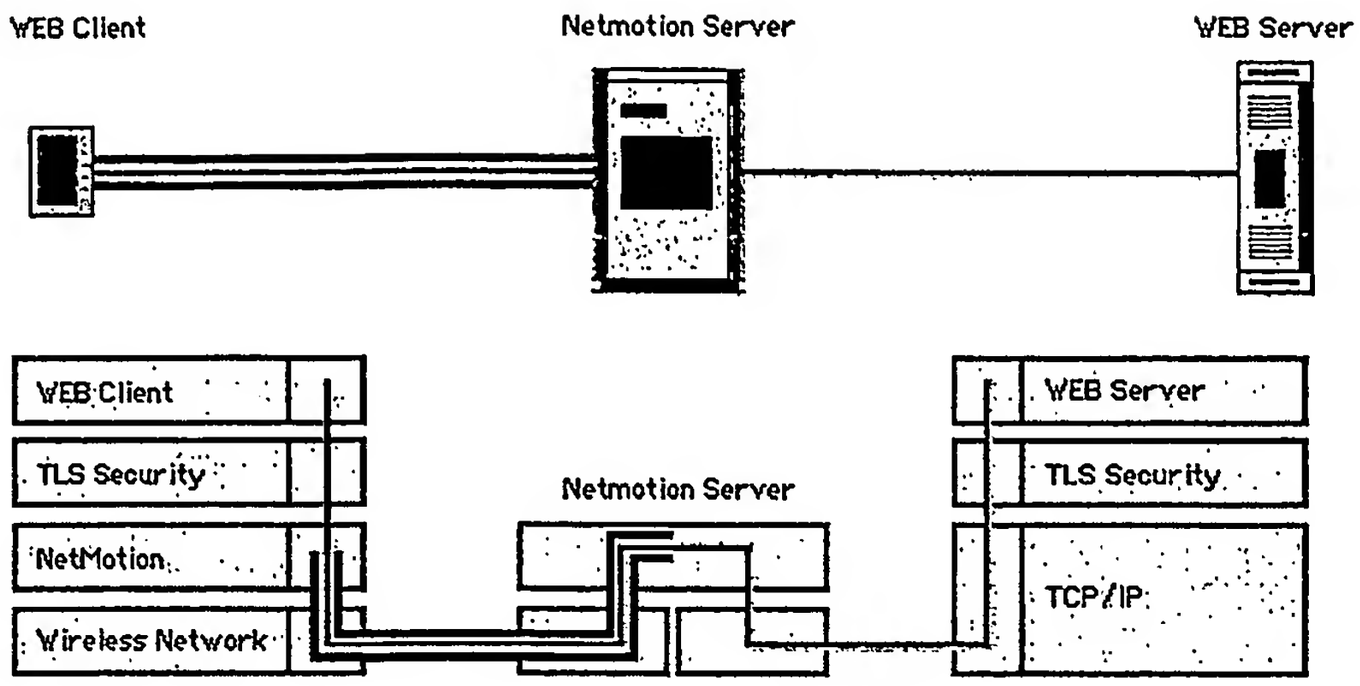

WEB Client
Netmotion Server
WEB Server
WEB Client
TLS Security
NetMotion
Wireless Network
Netmotion Server
WEB Server
TLS Security
TCP/IP
TLS Protected Data
NetMotion Services

Figure 5F

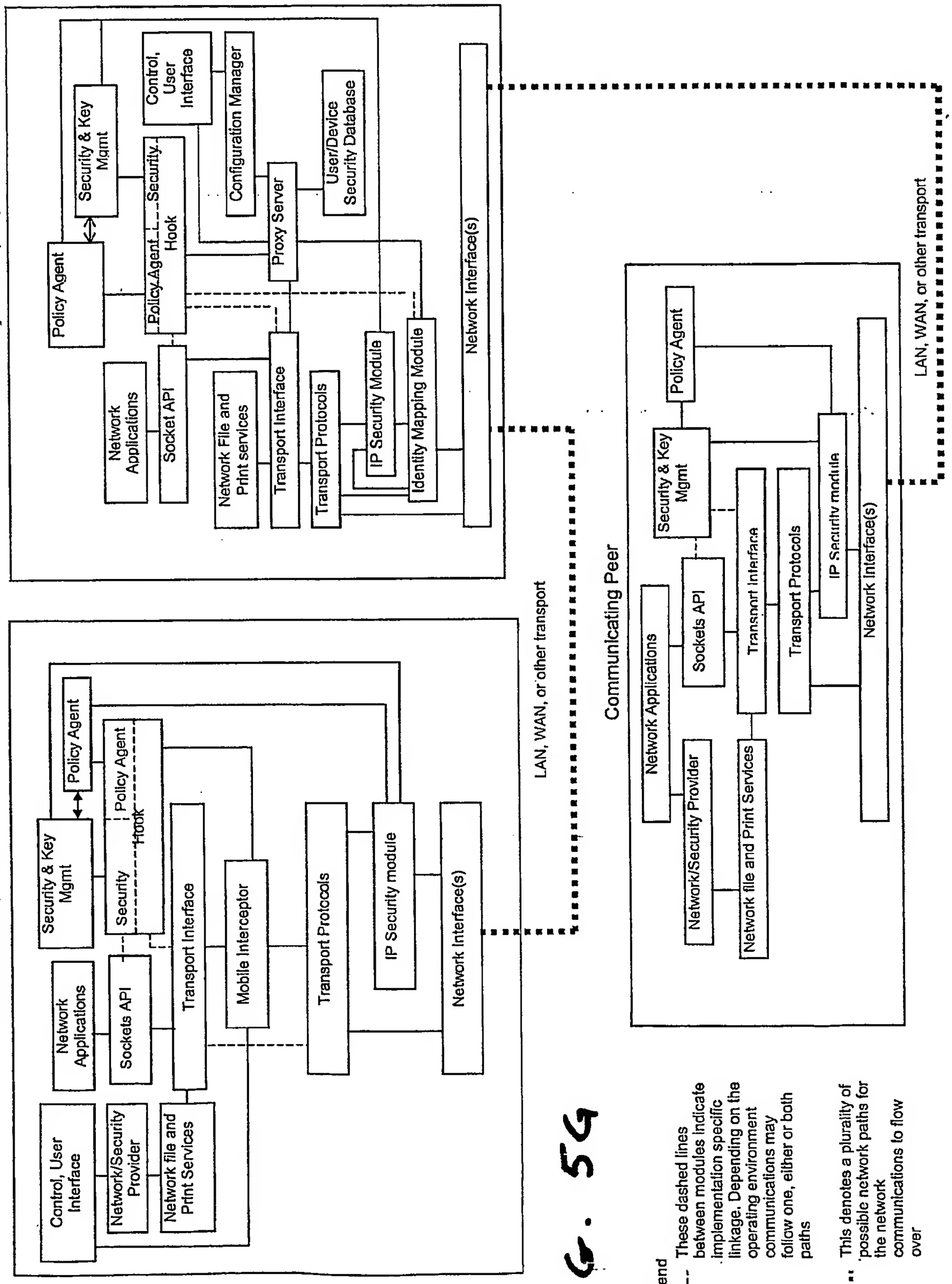


FIG. 5G

Legend

----- These dashed lines between modules indicate implementation specific linkage. Depending on the operating environment communications may follow one, either or both paths

..... This denotes a plurality of possible network paths for the network communications to flow over

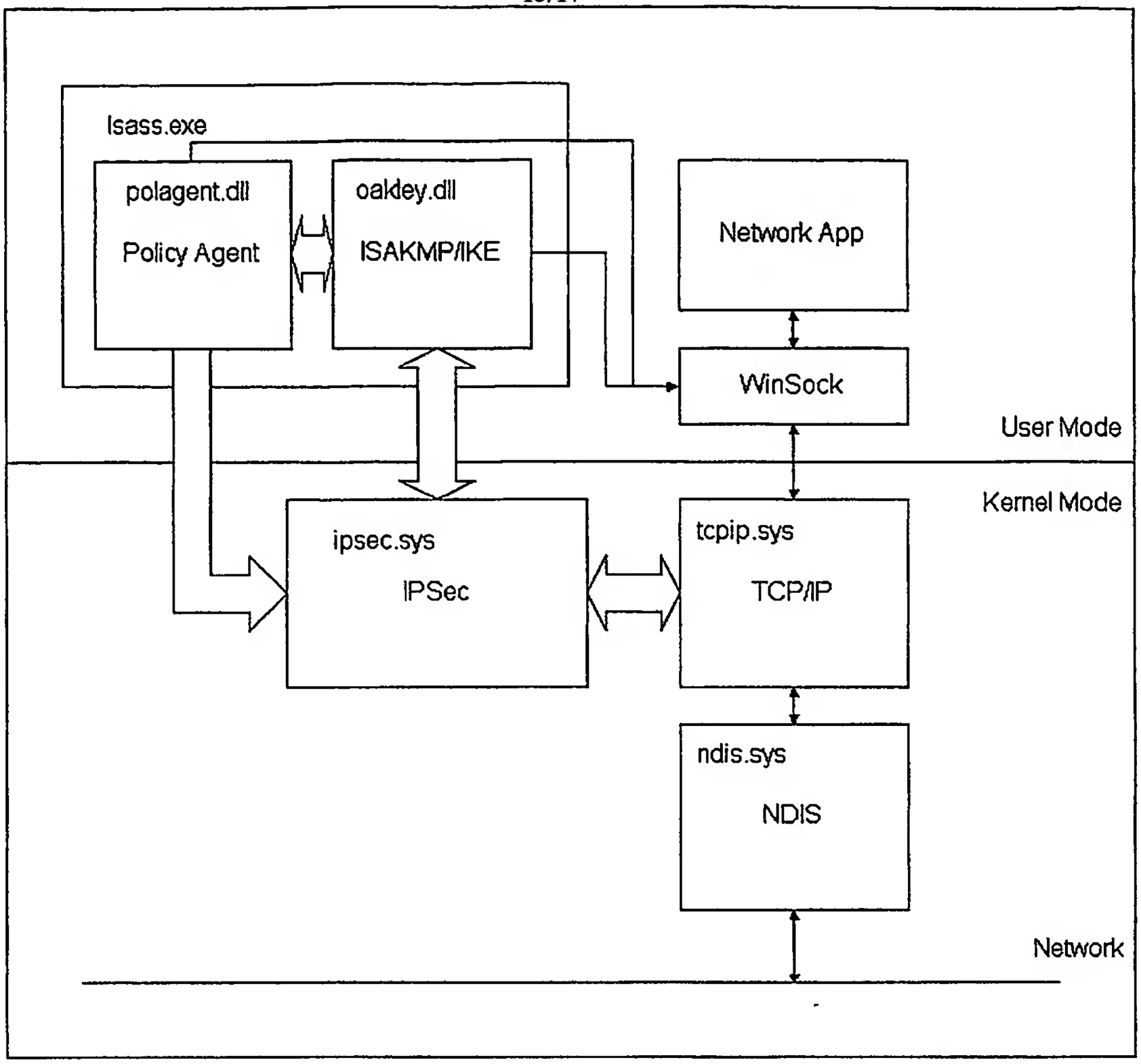


Figure 6 – Example IPSec Operating System Security Architecture
(Prior Art)

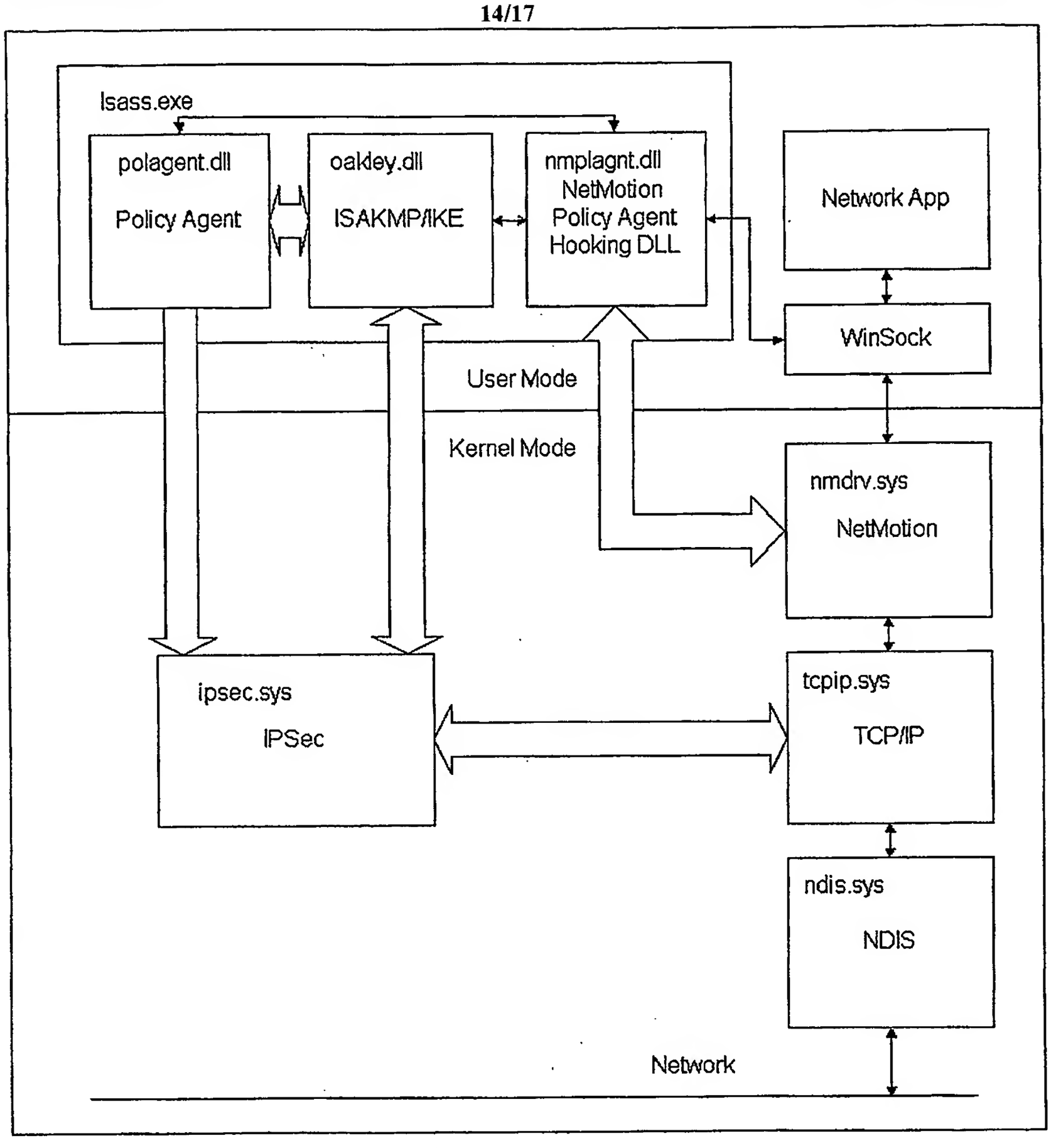


Figure 7 – Example Client Architecture

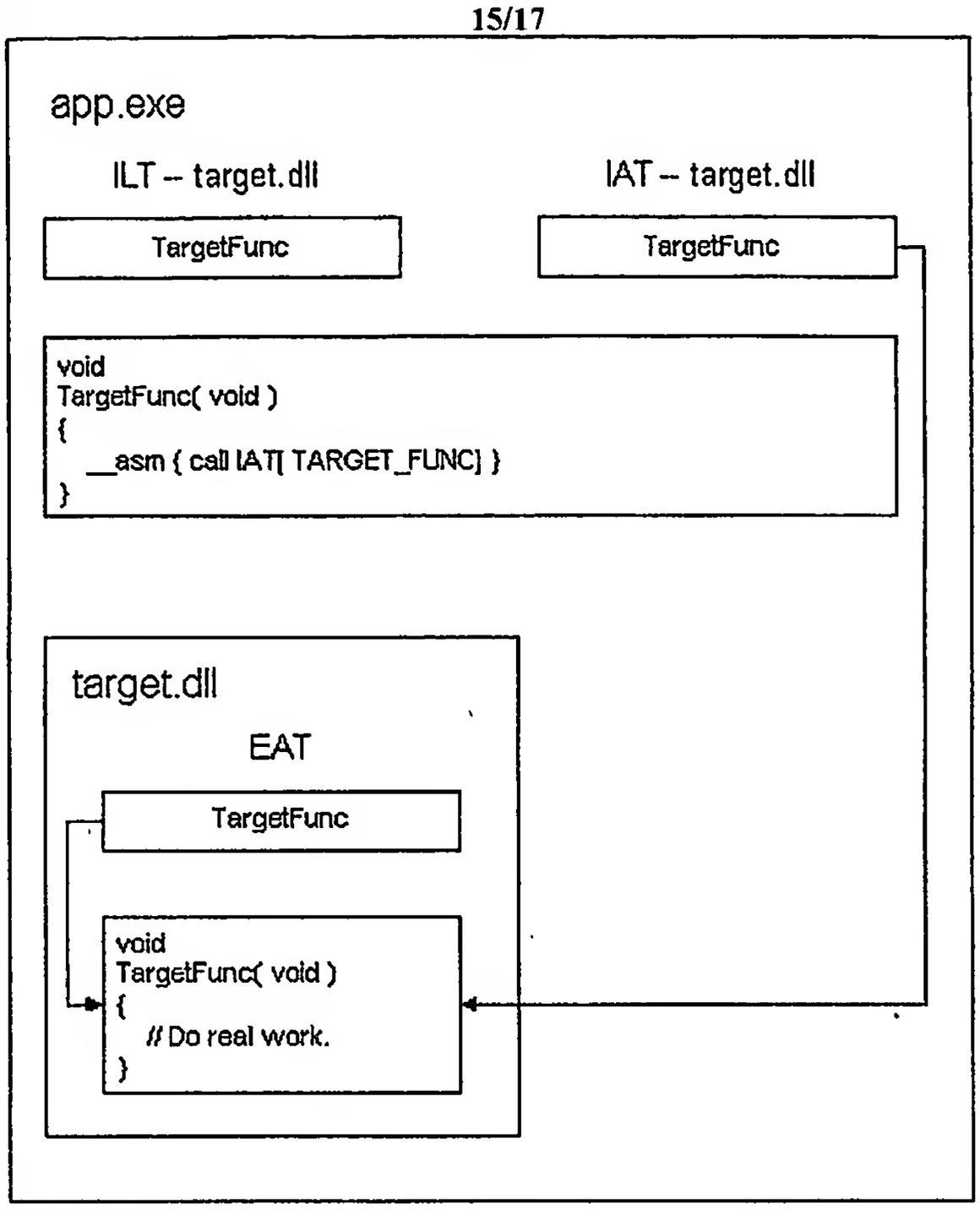


Figure 8 – Example Run-time Linking Sample

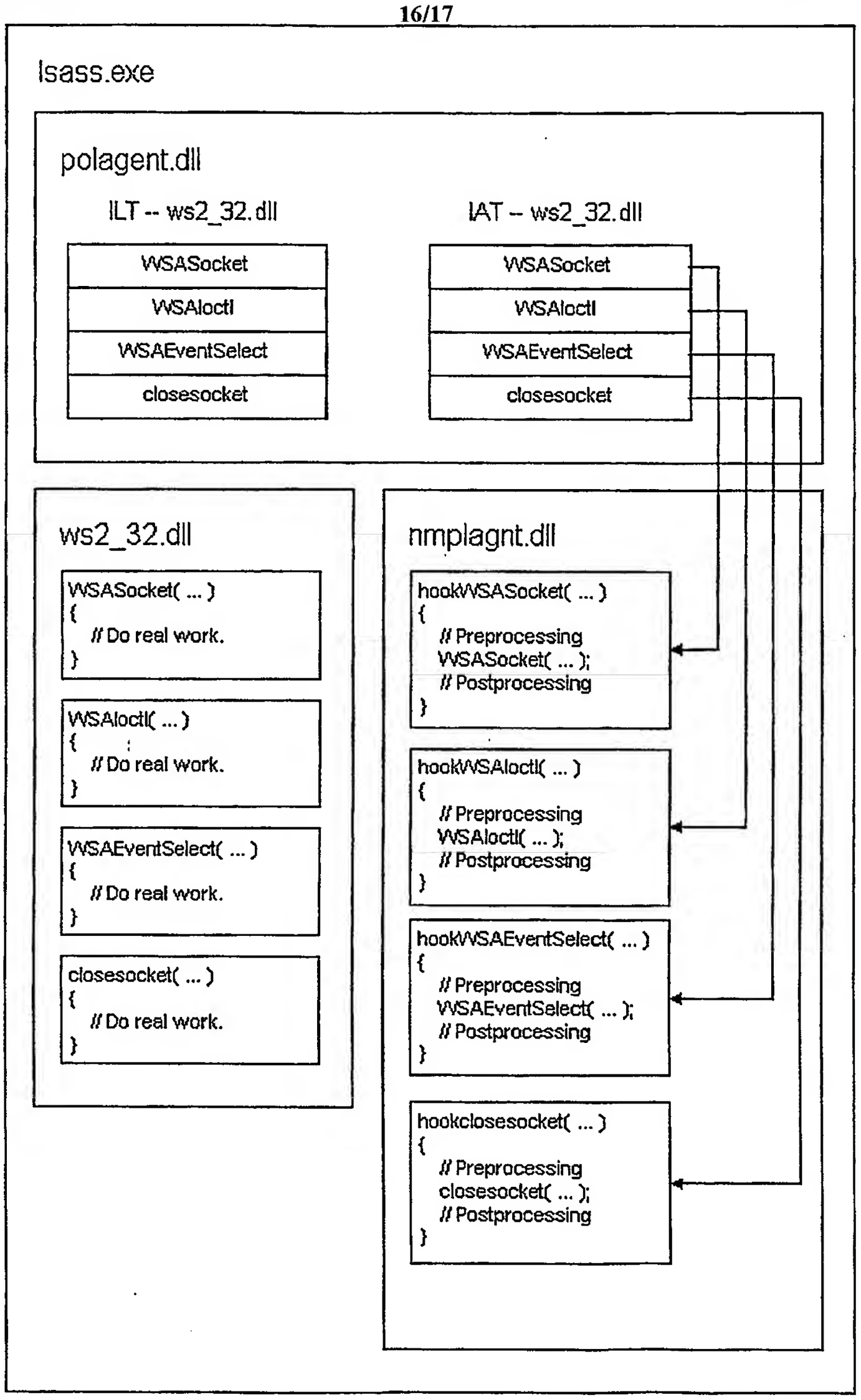


Figure 9 – Example Client Policy Agent Hooking

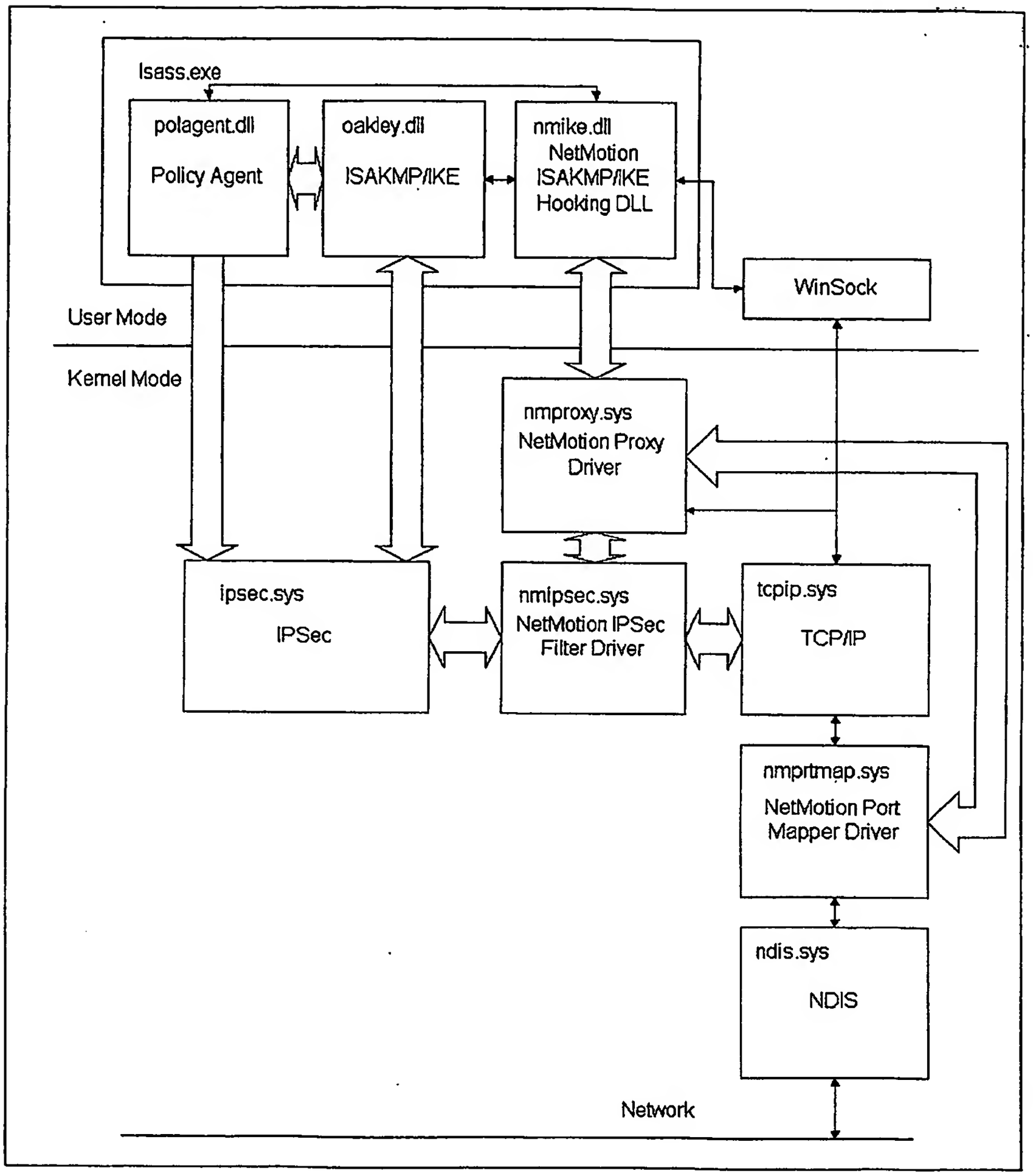


Figure 10 – Example Server Architecture

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US03/00817

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : H04L 9/00
US CL : 713/201

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 713/201

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2001/0009025 A1 (AHONEN) 19 July 2001 (19.07.2001), page 1, [0003], [[0010]-[0012], page 2, [0021], [0026], [0035]-[0048],page 3, [0059], [0070], page 4, [0083], page 5, [0096], page 6, [0124]-[0131], [0146]-[0147], page 7, [0153].	1-11, 30-40
Y	US 2001/0047474 A1 (TAKAGI et al.) 29 November 2001 (29.11.2001), pag 1, [0005]-[0013],page 2, [0017]-[0021], page 3, [0035]-[0040], page 4, [0041]-[0057], page 5, [0077]-[0078], page 6, [0083]-[0088], page 7, [0091]-[0099], page 8, [0102]-[0118], page 9, [0124]-[0126], [0129]-[0134], and page 10, claim5.	1-4, 6-11, 30-33, 35-40
Y	US 2001/0042201 A1 (YAMAGUCHI et al.) 15 November 2001 (15.11.2001), page 1, [0005]-[0009], page 2, [0016]-[0022], page 3, [0026], page 4, [0034]-[0045], page 5, [0065], [0072]-[0076], page 6, [0082]-[0086], page 7, [0091]-[0094], page 8, [0104], [0108]-[0112], page 9, [0117]	1-2, 6-11, 30-33, 35-40
Y,P	US 2002/0066036 A1 (MAKINENI et al.) 30 may 2002 (30.05.2002), abstract, page 1, [0003], [0004], [0009]-[0012], page 2, [0022]-[0028], page 3, [0029]-[0032], [0037]-[0039], and page 4, [0044]-[0045].	1-10, 30, 32, 34-39

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
04 April 2003 (04.04.2003)	21 APR 2003
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703)305-3230	Authorized officer for Gilberto Barron James R. Matthews Telephone No. 703-305-3900

Form PCT/ISA/210 (second sheet) (July 1998)

PCT/US03/00817

INTERNATIONAL SEARCH REPORT

C. (Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2001/0052081 A1 (MCKIBBEN et al.) 13 December.2001 (13.12.2001), Fig. 3-Fig.10 and page 1 through page 7.	1-11, 30-40
A,P	US 6,415,329 B1(GELMAN et al.) 02 July 2002 (02.07.2002), whole document.	1-11, 30-40
A	US 6,240,514 B1 (INOUE et al.) 29 May 2001 (29.05.2001), whole document.	1-11, 30-40

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US03/00817

Box I Observations where certain claims were found unsearchable (Continuation of Item 1 of first sheet)

This international report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claim Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claim Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claim Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of Item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:
Please See Continuation Sheet

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.: 1-11 and 30-40

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest.

☐ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

PCT/US03/00817

BOX II. OBSERVATIONS WHERE UNITY OF INVENTION IS LACKING

This application contains the following inventions or groups of inventions which are not so linked as to form a single general inventive concept under PCT Rule 13.1. In order for all inventions to be examined, the appropriate additional examination fees must be paid.

Group 1, claims 1-11 and 30-40, drawn to maintaining network communications which provides for the special technical features of detecting a network event and responding according to the nature of the event.

Group 2, claims 12-13 and 41-42, drawn to modifying an operating environment which provides for the special technical features of injecting computer instructions and redirecting the execution path.

Group 3, claims 14, 15-17, 43 and 44-46, drawn to providing a data communications in an environment which provides for the special technical features of virtualizing a network interface to shield network applications components and allowing other network to remain cognizant of occurrences in the network.

Group 4, claims 18-20 and 47-49, drawn to using plural IP security sessions which provides for the special technical features of distributing network communications and multiplexing/demultiplexing the communication flows.

Group 5, claims 21-22, 23, 50-51 and 52, drawn to a method which provides for the special technical features of facilitating (administering) the creation of plural IP security sessions, allowing, denying and/or delaying the communication flows and managing the policy concerning IP security sessions.

Group 6, claims 24-29 and 53-58, drawn to proxying mobile communications which provides for the special technical features of establishing communications between the mobile device and a peer and maintaining IP security sessions.

Group 7, claim 59 drawn to a storage medium which provides for the special technical features of containing two sets of instruction that inserts policy agent hooking a runtime module into an operating system and virtualizing driver into the operating system.

Group 8, claim 60, drawn to preparing a mobile device for secure communications which provides for the special technical features of downloading and executing sets of instructions onto the mobile device over a computer network.